

**PENERAPAN ALGORITMA *DIJKSTRA* PADA APLIKASI
PEKANBARU TAKSI GUIDE PADA PLATFORM ANDROID**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik
Pada Jurusan Teknik Informatika

oleh :

YONNI ARIS
10751000022



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2012**

KATA PENGANTAR



Alhamdulillah Robbil'alamin, penulis ucapkan syukur yang setinggi-tinggi ke-hadirat Allah SWT, karena atas segala limpahan rahmat dan karuniaNya yang diberikan sehingga penulis dapat menyelesaikan penelitian sekaligus penulisan laporan tugas akhir ini. *Allahumma sholli'ala Muhammad wa'ala ali sayyidina Muhammad*, yang tidak lupa penulis haturkan juga untuk junjungan alam, kekasih Allah, Rasul Allah, dan tauladan kita yakni Nabi Muhammad SAW.

Laporan tugas akhir ini merupakan salah satu prasyarat untuk memenuhi persyaratan akademis dalam rangka meraih gelar kesarjanaan di Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau (UIN SUSKA Riau). Selama menyelesaikan tugas akhir ini, penulis telah banyak mendapatkan bantuan, bimbingan, dan petunjuk dari banyak pihak baik secara langsung maupun tidak langsung. Untuk itu dalam kesempatan ini penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Prof. Dr. H. M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Dra. Yenita Morena, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau.
3. Ibu Dr. Okfalisa, ST. M.Sc, selaku Ketua Jurusan Teknik Informatika.
4. Bapak Febi Yanto, M.Kom selaku pembimbing dalam menyelesaikan tugas akhir ini. Terima kasih banyak atas dukungan-dukungan, nasehat, kebaikan-kebaikan, kesabaran dan semangat yang Bapak berikan kepada penulis.
5. Bapak Reski Mai Candra, ST, M.Sc, selaku Kordinator Tugas Akhir.

6. Bapak M. Irsyad, ST, M.T, selaku dosen penguji 1, terimakasih atas ilmu-ilmunya, saran-sarannya, perbaikan-perbaikannya, dan masukan yang Bapak berikan untuk penyempurnaan laporan ini.
7. Bapak Benny Sukma Negara, ST, M.T, selaku dosen penguji 2, terimakasih juga untuk ilmu-ilmunya, saran-sarannya, perbaikan-perbaikannya, dan masukan yang Bapak berikan untuk penyempurnaan laporan ini.
8. Dan terakhir, terimakasih pula penulis ucapkan untuk Almamater Teknik Informatika, Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim Riau serta pihak-pihak lain yang tidak dapat penulis sebutkan satu persatu. Terimakasih banyak atas bantuan dan dukungannya yang berharga.

Akhirnya, penulis menyadari dalam penulisan laporan ini masih terdapat kekurangan. Oleh karena itu, saran dan kritik sangat penulis harapkan untuk kemajuan penulis secara pribadi. Terimakasih.

Pekanbaru, 1 Juli 2013

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	iv
LEMBAR PERNYATAAN	v
LEMBAR PERSEMBAHAN	vi
<i>ABSTRACT</i>	vii
ABSTRAK	viii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN.....	xviii
DAFTAR TABEL.....	xix
DAFTAR SIMBOL.....	xxii
DAFTAR ISTILAH	xxiii
BAB I PENDAHULUAN	I-1
1.1. Latar Belakang	I-1
1.2. Rumusan Masalah	I-3
1.3. Batasan Penelitian	I-3
1.4. Tujuan Penelitian	I-3
1.5. Manfaat Penelitian	I-3
1.6. Sistematika Penulisan	I-4
BAB II LANDASAN TEORI.....	II-1
2.1. Transportasi di Pekanbaru	II-1
2.2. Sistem Informasi Geografis.....	II-2
2.3. Android.....	II-3
2.3.1. Sejarah Singkat.....	II-4

2.3.2. Arsitektur Android.....	II-6
2.4. <i>Location Based Service</i>	II-7
2.4.1. Komponen LBS	II-8
2.5. Teori Graf.....	II-10
2.5.1. Macam-Macam Graf.....	II-11
2.6. Algoritma <i>Dijkstra</i>	II-12
2.7. Metode Penelitian.....	II-14
2.7.1. Waterfall.....	II-14
2.7.2. <i>Rational Unified Process</i> (RUP)	II-16
2.4.1. Pengertian RUP	II-16
2.4.2. Fase RUP	II-18
BAB III METODOLOGI PENELITIAN.....	III-1
3.1. Tahapan <i>Waterfall</i>	III-1
3.1.1. Alur Tahapan <i>Waterfall</i>	III-1
BAB IV ANALISA DAN PERANCANGAN.....	IV-1
4.1. Analisa.....	IV-1
4.1.1. Defenisi Kebutuhan Sistem	IV-2
4.1.1.1. <i>Project Plan</i>	IV-3
A. Gambaran Umum Sistem.....	IV-3
B. Deskripsi Kebutuhan Sistem.....	IV-5
C. Fungsi Sistem.....	IV-8
4.1.1.2. Pembuatan Model.....	IV-9
A. Model Sistem	IV-9
B. Deskripsi Pengguna	IV-10
C. Pemodelan UML.....	IV-10
4.2. Perancangan	IV-17
4.2.1. Desain Sistem dan Aplikasi	IV-17
4.2.1.1. Perancangan Aplikasi.....	IV-17
A. Perancangan Struktur Menu Sistem.....	IV-18

B. Perancangan Antarmuka Pengguna Aplikasi	IV-18
4.2.1.2. Perancangan <i>Database</i>	IV-19
A. Perancangan <i>Database</i> Aplikasi	IV-19
B. Perancangan Anatmuka <i>Database</i> Aplikasi.....	IV-21
BAB V IMPLEMENTASI DAN PENGUJIAN	V-1
5.1. Implementasi	V-1
5.1.1. Pembuatan Aplikasi	V-1
5.1.1.1. Pengkodean	V-1
A. Pembuatan Aplikasi	V-1
B. Batasan Dalam Pembuatan Aplikasi	V-3
5.1.1.2. Implementasi	V-4
A. Hasil Implementasi Perhitungan Algoritma	V-4
B. Hasil Pengujian Algoritma Pada Beberapa	
Bentuk Graf	V-9
C. Imlementasi Fitur Pemberitahuan Aplikasi	V-13
D. Hasil Impmentasi	V-16
5.2. Pengujian	V-20
5.2.1. <i>Testing</i> dan Pengujian Aplikasi	V-20
5.2.1.1. Pengujian Aplikasi	V-20
A. Pengujian <i>Blackbox</i> Aplikasi	V-20
B. Pengujian Akses Aplikasi	V-23
C. Pengujian Fitur Pemberitahuan Aplikasi	V-28
5.2.2. Kesimpulan Pengujian	V-29
BAB VI PENUTUP	VI-1
6.1. Kesimpulan.....	VI-1
6.2. Saran.....	VI-2
DAFTAR PUSTAKA	
LAMPIRAN	
RIWAYAT HIDUP	

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan Pekanbaru sebagai ibukota propinsi Riau dapat dirasakan melalui berkembangnya akses jalan yang ada untuk mendukung pertumbuhan ekonomi. Mulai dari jalan-jalan protokol, hingga ke jalan alternatif. Berkembangnya akses jalan ini memberikan dampak positif terhadap akses transportasi melalui pilihan jalan yang mampu menghemat waktu serta biaya perjalanan, terutama pada angkutan umum taksi.

Keberadaan taksi di Pekanbaru sudah mulai banyak. Dimana terdapat 4 armada besar yaitu PUSKOPAU dengan taksi berwarna biru muda, KOPSI dengan taksi berwarna kuning dan RIAU TAKSI dengan taksi berwarna *silver metalik*, serta *Blue Bird* yang merupakan armada taksi yang baru yang bergejolak di Jakarta dengan warna biru pekat.

Perkembangan akses jalan serta keberadaan angkutan taksi di Kota Pekanbaru masih belum didukung dengan akses informasi jalan serta akses informasi transportasi yang bisa didapatkan dengan mudah, yang menyebabkan tidak semua masyarakat mengetahui semua jalan yang ada bahkan tidak sedikit dari mereka yang tidak mengetahui rute perjalanan yang akan ditempuh. Untuk itu dibutuhkan suatu teknologi yang dapat memberikan kemudahan dalam akses informasi jalan serta akses informasi transportasi di Kota Pekanbaru yang bisa diakses dimana saja, kapan saja dan oleh siapa saja.

Salah satu teknologi yang bisa digunakan adalah *Location Based Service* (LBS) yang mampu menyediakan layanan berbasis lokasi kepada pengguna *mobile smartphone* yang menerapkan sistem *Global Positioning Satellite* (GPS).

Teknologi ini mampu menunjukkan lokasi dimana pengguna berada melalui map *street* yang tersedia.

Melalui teknologi LBS ini, maka perlu dikembangkan sebuah aplikasi yang mampu menyediakan informasi jalan untuk Kota Pekanbaru yang dapat menunjukkan rute terpendek perjalanan serta memberikan perkiraan biaya perjalanan pada angkutan taksi untuk calon penumpang.

Untuk menentukan rute terpendek pada aplikasi ini, diterapkan Algoritma *Dijkstra*. Dari beberapa penelitian yang telah dilakukan sebelumnya terkait perbandingan algoritma penentuan jalur terpendek, algoritma *Dijkstra* memiliki kecepatan yang lebih cepat dalam memproses daripada algoritma yang lain seperti algoritma A* (*A star*), Algoritma Semut, dan Algoritma *Bellman-Ford*. Kesimpulan ini didapat melalui penelitian yang dilakukan oleh Michell Setyawati Handaka pada tahun 2010 dalam penelitiannya “*Perbandingan Algoritma Dijkstra (Greedy), Bellman-Ford (BFS-DFS), dan Floyd-Warshall (Dynamic Programming) dalam Pengaplikasian Lintasan Terpendek pada Link-State Routing Protocol*”. Michell menarik kesimpulan bahwa algoritma *Dijkstra* dapat menjadi pilihan favorit dari pada algoritma *shortest path* yang lain, mengingat kebutuhan waktu dan ruangnya sangat kecil.

Selain penelitian mengenai perbandingan *shortest path algorithm*, ada beberapa penelitian yang pernah dilakukan sebelumnya yaitu dilakukan oleh Yulia Kartika Gunadi dan Jeffrey Tanuhardja pada tahun 2002 yang membahas Perencanaan Rute Perjalanan Di Jawa Timur Dengan Dukungan GIS Menggunakan Metode Dijkstra serta penelitian yang dilakukan oleh Luh Joni Erawati Dewi pada tahun 2010 yang membahas tentang Pencarian Rute Terpendek Tempat Wisata Di Bali Dengan Menggunakan Algoritma *Dijkstra*. Kedua penelitian ini membuat suatu sistem informasi yang berbasis *desktop* yang memberikan rute terpendek untuk mencapai tujuan. Hingga sekarang ini, belum ada penelitian yang membahas mengenai pencarian rute terpendek menggunakan angkutan umum taksi yang mampu memperkirakan biaya perjalanan, waktu tempuh, jarak tempuh, serta mampu untuk memberikan solusi jalur alternatif apabila terjadi kemacetan dengan studi kasus di Kota Pekanbaru.

1.2. Rumusan Masalah

Dari latar belakang yang telah dipaparkan sebelumnya, dapat diambil rumusan masalah yaitu "bagaimana membangun dan merancang sebuah aplikasi yang mampu memberikan informasi rute terpendek serta perkiraan biaya perjalanan pada angkutan taksi untuk wilayah Kota Pekanbaru dengan menerapkan algoritma *dijkstra*".

1.3. Batasan Penelitian

Untuk mendapatkan hasil yang optimal dalam penelitian tugas akhir ini, maka diberikan batasan-batasan penelitian sebagai berikut:

- a. Karena rute terpendek akan menghasilkan *cost* yang rendah, maka transportasi umum yang digunakan adalah taksi. Namun aplikasi ini juga bisa digunakan oleh pemakai kendaraan pribadi tanpa memperhitungkan *cost* yang dikeluarkan.
- b. Terdapat *update* kondisi jalan lalu lintas untuk memudahkan pengguna agar tidak terjebak macet atau terjadinya pengalihan jalan.
- c. Tipe jalan dibatasi hanya pada jalan kolektor atau jalan protokol.
- d. Data lokasi tujuan dibagi menjadi 6 kategori, yaitu hotel, rumah sakit, kuliner, tempat rekreasi, *mall*, dan *public transport*.

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai penulis dari penelitian Tugas Akhir ini adalah membangun sebuah aplikasi yang mampu memberikan informasi jalan untuk wilayah Kota Pekanbaru yang berjalan di *platform Android*, yang mampu menunjukkan jalur terdekat yang akan dilalui, mampu memberikan perkiraan biaya untuk sampai ke tempat tujuan, memberikan informasi jarak tempuh yang disertai waktu tempuh yang akan dilalui, mampu memberikan informasi perkiraan titik kemacetan sehingga pengguna bisa menghindarinya serta dilengkapi dengan peringatan (*alert*) apabila posisi pengguna tidak berada pada jalur yang telah ditentukan sebelumnya pada angkutan taksi.

1.5. Manfaat Penelitian

Beberapa manfaat dari penelitian tentang aplikasi Pekanbaru Taksi *Guide* ini adalah:

- a. Dapat memberikan informasi mengenai rute perjalanan yang ada di Pekanbaru.
- b. Memberikan kemudahan dalam akses informasi transportasi di Kota Pekanbaru, khususnya taksi. Akses informasi tersebut berupa perkiraan biaya perjalanan, jarak tempuh, dan lama perjalanan.
- c. Membantu pengguna untuk mengetahui kondisi keadaan lalu lintas terkini di jalan sehingga menghindari pengguna dari kemacetan atau pengalihan jalan yang disebabkan oleh suatu kejadian atau peristiwa.

1.6. Sistematika Penulisan

Berikut merupakan rencana susunan sistematika penulisan laporan Tugas Akhir yang akan dibuat :

BAB I Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan sistematika penulisan dari Tugas Akhir yang akan dibuat.

BAB II Landasan Teori

Bab ini membahas tentang teori-teori umum dan khusus yang berhubungan dengan Tugas Akhir ini.

BAB III Metodologi Penelitian

Bab ini membahas langkah-langkah yang dilaksanakan dalam proses penelitian, yaitu pengamatan pendahuluan, tahapan identifikasi masalah, perumusan masalah, analisa aplikasi, perancangan aplikasi dan implementasi aplikasi beserta pengujian.

BAB IV Analisa dan Perancangan

Bab ini berisi pembahasan mengenai kebutuhan sistem, yang terdiri dari : *Flowchart* sistem, *Unified Modeling Language* (UML), *User interface*, perancangan menggunakan pendekatan berorientasi objek.

BAB V Implementasi dan Pengujian

Bab ini berisi penjelasan mengenai implementasi dan pengujian dari aplikasi yang terdiri dari: batasan implementasi, lingkungan implementasi, hasil implementasi, pengujian aplikasi dan kesimpulan dari pengujian

BAB VI Kesimpulan dan Saran

Bagian ini berisi kesimpulan yang dihasilkan dari pembahasan dan pengembangan dari aplikasi *transportation guide* untuk wilayah Kota Pekanbaru.

BAB II

LANDASAN TEORI

2.1. Transportasi di Pekanbaru

Transportasi diartikan sebagai pemindahan barang dan manusia dari tempat asal ke tempat tujuan. Dalam hubungan ini dapat terlihat tiga hal berikut: adanya muatan yang diangkut; tersedianya kendaraan sebagai alat angkutan dan tersedianya jalan yang dapat dilalui oleh alat angkutan. Proses pemindahan (transportasi) merupakan gerakan dari tempat asal, dimana kegiatan pengangkutan dimulai, ke tempat tujuan, dimana kegiatan diakhiri. Transportasi berfungsi sebagai sektor penunjang ekonomi (*the promoting sector*) dan pemberi jasa (*the servicing sector*) bagi perkembangan ekonomi nasional (Nasution H.M.N, 1996).

Untuk Kota Pekanbaru sendiri yang merupakan ibukota dari Propinsi Riau, memiliki 3 jenis transportasi layaknya kota-kota besar yang lain, yaitu transportasi air yang berpusat di Pelabuhan Sei Duku yang menyediakan kapal-kapal penyebrangan, transportasi udara yang berpusat di Bandara Sultan Syarif Kasim 2, dan transportasi darat yang meliputi angkutan umum, bus, dan taksi.

Terdapat 4 jenis transportasi darat di Kota Bertuah (julukan Kota Pekanbaru), yaitu angkutan umum yang dikenal dengan oplet, bus kota, bus *Trans Metro Pekanbaru*, dan juga taksi.

Oplet atau angkutan umum sebagai angkutan masal memiliki banyak jalur trayek serta warna yang berbeda-beda untuk tiap trayeknya. Mulai dari trayek Harapan Raya-Kulim, Panam-Sukajadi, Tangkerang-Pasar Locket, dan Tangkerang-Pasar Pusat. Umumnya, jalur trayek angkutan umum atau lebih dikenal dengan sebutan oplet bagi masyarakat Pekanbaru ini, melewati jalan-jalan

protokol di kota pekanbaru. Seperti jalan H. Imam Munandar, Jalan Tuanku Tambusai, Jalan Sudirman, dan Jalan HR Subrantas.

Transportasi jenis bus ada 2, yaitu bus kota dan Bus Trans Metro Pekanbaru. Perbedaan dari keduanya adalah Bus Trans Metro Pekanbaru lebih nyaman dengan dilengkapi fasilitas ac serta berhenti pada halte-halte yang telah ditentukan sedangkan pada bus kota sebaliknya.

Sedangkan untuk angkutan modern yaitu taksi, terdapat 3 armada yang beroperasi. Yaitu Taksi Puskopau (biru), Taksi Kopsi (Kuning), dan Riau Taksi (abu-abu atau *silver*). Ketiga armada taksi ini saling bersaing memberikan pelayanan yang terbaik kepada penumpang.

Untuk tarif dasar, seluruh armada taksi berpedoman pada SK Walikota Pekanbaru Nomor 31/2009 tanggal 24 Juli 2009 tentang pemberlakuan tarif dasar taksi yang baru yang ditunjukkan pada tabel 2.1 dibawah ini:

Tabel 2.1 Tarif Dasar Taksi Kota Pekanbaru

Jenis Tarif Dasar	Biaya (Rp)
Tarif buka pintu	6.000
Tarif perkilometer	3.500
Tarif minimal argo	20.000
Tarif pembatalan sepihak	10.000
Tarif tunggu per jam	20.000

2.2. Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) merupakan suatu kesatuan formal yang terdiri dari berbagai sumber daya fisik dan logika yang berkenaan dengan objek-objek yang terdapat di permukaan bumi. Dan SIG merupakan sejenis perangkat lunak atau aplikasi yang dapat digunakan untuk pemasukan data, penyimpanan data, manipulasi data, menampilkan data, dan keluaran informasi geografis (Adeline dan Eri, 2007).

Teknologi GIS mengintegrasikan operasi pengolahan data berbasis *database* yang biasa digunakan saat ini, seperti pengambilan data berdasarkan

kebutuhan, serta analisis *statistic* dengan menggunakan visualisasi yang khas serta berbagai keuntungan yang mampu ditawarkan melalui analisis geografis melalui gambar-gambar petanya.

Sistem Informasi Geografis terdiri dari 4 komponen:

a. Perangkat keras

Seperti komputer untuk penginputan dan pengolahan data, dan perangkat keras untuk keluaran data seperti *printer*, *plotter* dan lain sebagainya.

b. Perangkat Lunak

Merupakan sistem aplikasi yang tersusun secara modular dengan basis data sebagai pemegang peranan penting dalam sistem perangkat lunak. Lebih tepatnya berupa tools-tools untuk membuat dan mengolah system informasi geografis. Seperti *Arc View*, *Idrisi*, *ARC/INFO*, *ILWIS*, *MapInfo* dan lain lain.

c. Data dan Informasi Geografis

SIG dapat melakukan pengumpulan data, penyimpanan data, dan pemrosesan data dan informasi yang diperlukan baik secara tidak langsung dengan cara mengimpornya dari perangkat lunak SIG yang lain, maupun secara langsung dengan cara mendijitasi data spasial dari peta.

d. Manajemen

Keberhasilan suatu sistem informasi geografis akan berhasil jika diatur sedemikian rupa dan dikerjakan oleh orang-orang yang memiliki keahlian pada bidangnya (Adeline dan Eri, 2007).

2.3. Android

Android merupakan sebuah sistem operasi untuk perangkat *mobile*, yang dibangun diatas *Kernel Linux* versi 2.6, dimana *kernel* ini telah teruji ketangguhannya di berbagai distro *Linux*. OS ini merupakan generasi baru *platform mobile*, *Platform* ini bersifat *open source* sehingga siapapun bebas

mengubah dan mengembangkan OS ini sesuai kebutuhan. *Android* bertujuan untuk memajukan piranti lunak *mobile* agar setiap pengguna bisa mendapatkan pengalaman baru serta menggali potensi yang ada *platform* ini (Neven Vr ek, 2009).

Pada masa sekarang ini, android merupakan sistem operasi *mobile* yang sangat berkembang. Hal ini merupakan dampak positif dari sifatnya yang *open source* sehingga memberikan kemudahan bagi siapa saja yang ingin mengembangkannya.

2.3.1. Sejarah Singkat

Pada tanggal 22 oktober 2008, HTC *Dream* adalah *smartphone* pertama yang menggunakan sistem operasi *android*. Dan hingga sekarang ini, telah banyak vendor-vendor *smartphone* yang menggunakan sistem operasi ini.

Adapun versi-versi *android* yang pernah dirilis oleh *Google.inc* adalah sebagai berikut:

a. *Android* versi 1.1

Rilis pada maret 2009. *Android* versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam *alarm*, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan *email*.

b. *Android* versi 1.5 (*Cupcake*)

Rilis pada pertengahan Mei 2009. Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam *seluler* versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan *Bluetooth A2DP*, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

c. *Android* versi 1.6 (*Donut*)

Rilis pada pertengahan September 2009. menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang

memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, *camcorder* dan galeri yang diintegrasikan; CDMA / EVDO, 802.1x, VPN, *Gestures*, dan *Text-to-speech engine*; kemampuan dial kontak; teknologi *text to change speech* (tidak tersedia pada semua ponsel) dan pengadaan resolusi VWGA.

d. *Android* versi 2.0/2.1 (*eclair*)

Rilis pada 3 desember 2009. pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan *Bluetooth* 2.1.

e. *Android* versi 2.2 (*Frozen Yoghurt / Froyo*)

Dukungan *Adobe Flash* 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi *V8 JavaScript engine* yang dipakai *Google Chrome* yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam SD Card, kemampuan *WiFi Hotspot portabel*, dan kemampuan *auto update* dalam aplikasi *Android Market*.

f. *Android* versi 2.3 (*Gingerbread*)

Peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (*reverb*, *equalization*, *headphone virtualization*, dan *bass boost*), dukungan kemampuan *Near Field Communication* (NFC) (Syafaat, 2011)

g. *Android* versi 3.0/3.1 (*Honeycomb*)

Android Honeycomb dirancang khusus untuk *tablet*. *Android* versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk *tablet*. *Honeycomb* juga mendukung *multi prosesor* dan juga akselerasi perangkat keras (*hardware*) untuk grafis

h. *Android* versi 4.0 (ICS :*Ice Cream Sandwich*)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara *offline*, dan berbagi informasi dengan menggunakan NFC.

i. *Android* versi 4.1 dan 4.2 (JB : *Jelly Bean*)

Android Jelly Bean versi 4.1 diumumkan pada tanggal 27 Juni 2012. Perubahan yang terjadi yaitu meningkatkan kemudahan dalam penggunaan sistem operasi karena lebih *fast and smooth*, keindahan dari tampilan, perubahan pada notifikasi bar yang dibuat lebih dinamis dan bisa diatur oleh pengguna, dan *widget* cerdas yang bisa ukurannya bisa diatur sesuai dengan kondisi layar.

Android Jelly Bean versi 4.2 diumumkan 3 bulan setelah versi sebelumnya diumumkan. Pada versi ini perubahan yang terjadi yaitu peningkatan kecepatan dalam melakukan *multitasking*, serta penambahan fitur baru seperti *Photo Sphere*, desain baru dari aplikasi kamera, *keyboard gesture typing*, dan *Google Now* (<http://www.android.com>).

2.3.2. *Arsitektur Android*

Secara garis besarnya, arsitektur dari sistem operasi *mobile android* ini dijelaskan sebagai berikut :

a. *Applications dan Widgets*

Layer ini merupakan daerah dimana si pengguna berhubungan dengan aplikasi yang digunakan, dari mengunduh (*download*), mengunggah (*install*) hingga menjalankan aplikasi tersebut. Pada layer ini, semua aplikasi yang digunakan menggunakan bahasa pemrograman *Java*.

b. Applications Frameworks

Adalah layer dimana para pembuat aplikasi melakukan pengembangan ataupun pembuatan aplikasi baru. karena pada layer ini, diberikan akses penuh terhadap sistem operasi *android*.

c. Libraries

Merupakan tempat dimana fitur-fitur *android* berada. Beberapa fitur tersebut adalah:

1. *Libraries media*, berhubungan dengan video dan audio
2. *Libraries* untuk mengatur tampilan layar
3. *Libraries graphics*
4. *Libraries SQLite* untuk database
5. *Libraries SSL* dan *Webkit* yang terintegrasi dengan *browser* dan keamanan
6. *Libraries LiveWebcore*
7. *Libraries 3D*

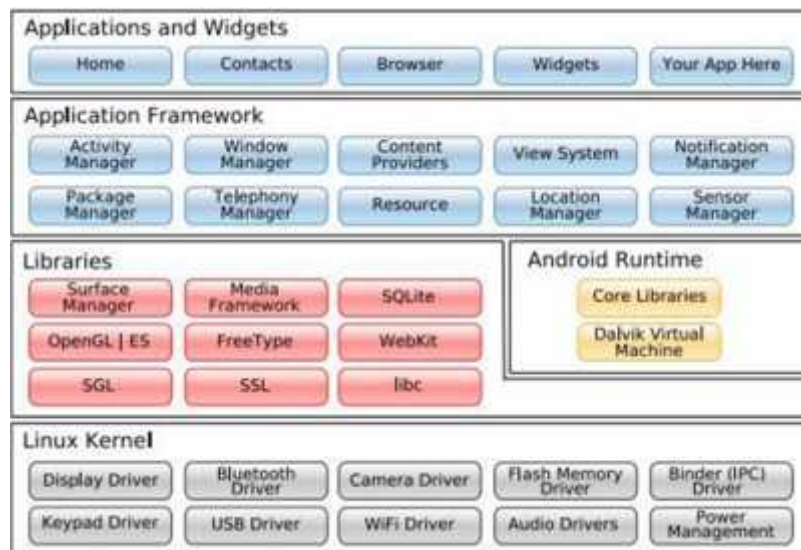
d. Android Run Time

Layer yang mampu menjalankan aplikasi *android* yang telah diinstal. Didalam layer ini terbagi 2 yaitu:

1. *Core Libraries*
2. *Davik Virtual Machine (DVM)*

e. Linux kernel

Layer ini merupakan inti dari sistem operasi *android*. Berisi file-file sistem yang mampu mengatur sistem *processing*, *memory*, *drivers*, dan sistem yang lainnya pada *android*. *Kernel* yang digunakan pada sistem operasi *android* ini adalah *linux kernel release 2.6* (Syafaat, 2011).



Gambar 2.1 Arsitektur Android

(sumber : <http://www.android-indonesia.com/home/15-developers/12156-arsitektur-android>)

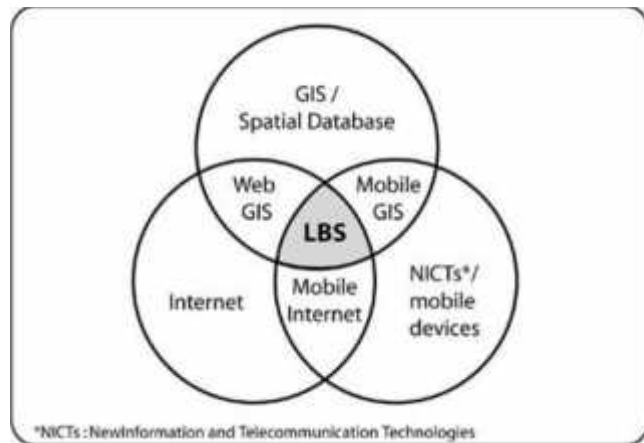
2.4. Location Based Service

Location Based Service atau lebih dikenal dengan layanan berbasis lokasi merupakan suatu teknologi yang digunakan untuk menemukan suatu lokasi yang berbasis map pada perangkat yang digunakan. Perangkat tersebut biasa berupa PDA, atau *Smartphone* yang telah memiliki sistem GPS didalamnya.

Sekarang ini, pengembangan LBS lebih banyak ke ponsel-ponsel cerdas. Ini dapat dilihat dengan berkembangnya aplikasi-aplikasi yang berbasis LBS pada perangkat mobile smartphone (Syafaat, 2011).

LBS memberikan kemungkinan komunikasi dan interaksi dua arah antara pengguna dan penyedia layanan,. Oleh karena itu pengguna bisa memberitahu penyedia layanan untuk mendapatkan informasi yang dibutuhkan, dengan referensi posisi pengguna tersebut.

Layanan berbasis lokasi dapat digambarkan sebagai suatu layanan yang berada pada pertemuan tiga teknologi yaitu : Geographic Information System, Internet Service, dan Mobile Devices, hal ini dapat dilihat pada gambar LBS adalah pertemuan dari tiga teknologi. (Juwita Imaniar, 2012)



Gambar 2.2 LBS sebagai simpang tiga teknologi
(sumber : Neven Vr ek, 2009)

2.4.1. Komponen LBS

Terdapat empat komponen pendukung utama dalam teknologi Layanan Berbasis Lokasi, antara lain (Steiniger, 2007):

1. Perangkat *Mobile*

Perangkat *Mobile* adalah komponen penting dalam LBS. Perangkat ini berfungsi sebagai alat bantu bagi pengguna untuk meminta informasi berupa teks, suara, gambar dan lain sebagainya.

2. Jaringan Komunikasi

Komponen ini berfungsi sebagai jalur penghubung untuk mengirimkan data. Pengiriman data diawali oleh *request* dari pengguna ke penyedia layanan dari perangkat *mobile*-nya, kemudian penyedia layanan mengirimkan data hasil dari *request* pengguna.

3. Komponen *Positioning* atau penunjuk lokasi

Setiap layanan yang diberikan oleh penyedia layanan biasanya akan berdasarkan pada posisi pengguna yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang berfungsi sebagai pengolah/pemroses yang akan menentukan posisi pengguna layanan saat itu. Posisi pengguna tersebut bisa didapatkan melalui jaringan komunikasi *mobile* atau juga menggunakan *Global Positioning Sistem* (GPS).

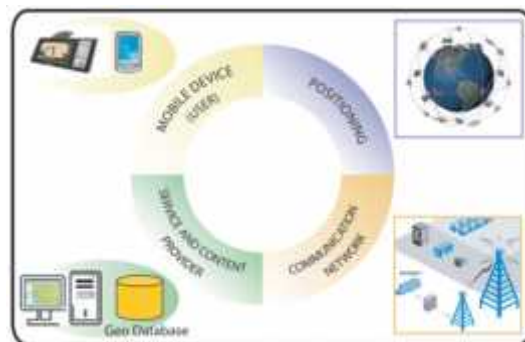
4. Penyedia layanan dan aplikasi

Penyedia layanan memberikan berbagai macam layanan yang bisa digunakan oleh pengguna. Sebagai contoh ketika pengguna meminta layanan agar bisa tahu posisinya saat itu, maka aplikasi dan penyedia layanan langsung memproses permintaan tersebut, dan mengirimkannya ke perangkat pengguna.

5. Penyedia data dan konten

Penyedia layanan tidak selalu menyimpan seluruh data dan informasi yang diolahnya. Karena bisa jadi berbagai macam data dan informasi yang diolah tersebut berasal dari pengembang/pihak ketiga yang memang memiliki otoritas untuk menyimpannya. Sebagai contoh basis data geografis dan lokasi bisa saja berasal dari badan-badan milik pemerintah atau juga data-data perusahaan/bisnis/industri bisa saja berasal dari *Yellow Pages*, maupun perusahaan penyedia data lainnya.

Secara lengkap keempat komponen pendukung LBS tersebut dapat dilihat pada gambar berikut.



Gambar 2.3 Komponen pendukung utama teknologi LBS
(sumber: Steiniger, 2007)

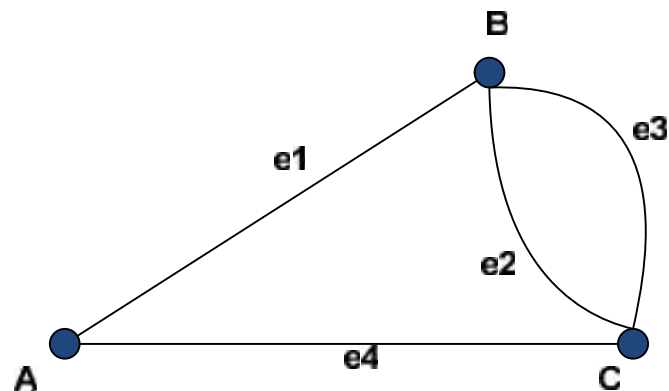
2.5. Teori Graf

Graf adalah kumpulan simpul (*nodes*) yang dihubungkan satu sama lain melalui sisi/busur (*edges* atau *arcs*). Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antar objek tersebut. Representasi visual dari graf adalah suatu objek dinyatakan sebagai noktah, bulatan atau titik, dan hubungan natar objek dinyatakan dengan garis. (Rinaldi Munir, 2005)

Suatu Graf G terdiri dari dua himpunan yaitu himpunan V dan himpunan E .

- Verteks* (simpul) dilambangkan dengan V = himpunan simpul yang terbatas dan tidak kosong.
- Edge* (sisi/busur) dilambangkan dengan E = himpunan busur yang menghubungkan sepasang simpul.

Dapat dikatakan graf adalah kumpulan dari simpul-simpul yang dihubungkan oleh sisi-sisi.



Gambar 2.1 Graf G
(sumber : Ifatul Faizah, 2010)

Pada G diatas, graf terdiri dari himpunan V dan E yaitu:

$$V = \{A, B, C\}$$

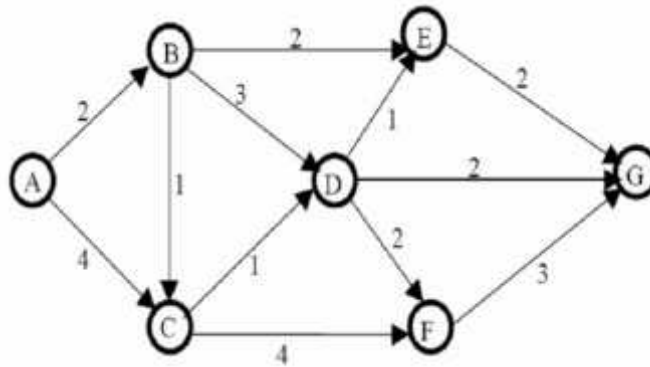
$$E = \{e1, e2, e3, e4\}$$

$$= \{(A,B),(B,C),(B,C),(A,C)\}$$

2.5.1. Macam-Macam Graf :

- Graf berarah dan berbobot

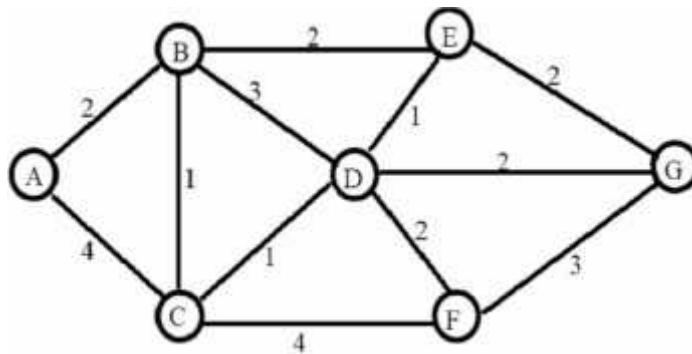
Adalah graf yang mempunyai bobot nilai yang menentukan nilai dari setiap edge disertai arahnya dari noktah satu menuju ke noktah yang lainnya.



Gambar 2.5. Graf berarah dan berbobot
(sumber : Ifatul Faizah, 2010)

- b. Graf tidak berarah dan berbobot

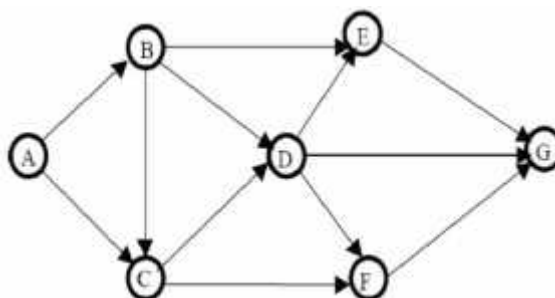
Graf ini hampir sama dengan graf sebelumnya, yang membedakannya adalah pada tiap *edge*, tidak memiliki arah baik dari noktah awal ke tujuan ataupun sebaliknya.



Gambar 2.6. Graf tidak berarah dan berbobot
(sumber : Ifatul Faizah, 2010)

- c. Graf berarah dan tidak berbobot

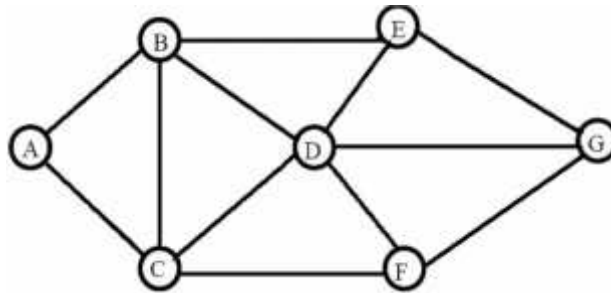
Tiap busur mempunyai anak panah yang saling menghubungkan noktah dan tidak memiliki berbobot nilai.



Gambar 2.7. Graf berarah dan tidak berbobot
(sumber : Ifatul Faizah, 2010)

- d. Graf tidak berarah dan tidak berbobot

Tiap busur tidak mempunyai anak panah dan tidak mempunyai bobot nilai.



Gambar 2.8. Graf tidak berarah dan tidak berbobot
(sumber : Ifatul Faizah, 2010)

2.6. Algoritma *Dijkstra*

Algoritma ini dinamakan sesuai dengan nama penemunya, yaitu seorang ilmuwan komputer berkebangsaan Belanda yang bernama Edsger Dijkstra. Algoritma Dijkstra termasuk kedalam pembahasan teori graf pada matematika diskrit yang berhubungan dengan graf berbobot dan lintasan terpendek (*shortest path*). Algoritma ini digunakan untuk mencari lintasan terpendek pada sebuah graf berarah. Cara kerja algoritma *Dijkstra* memakai strategi *greedy*, dimana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih (Rinaldi Munir, 2005).

Contoh penerapan algoritma *dijkstra* adalah pencarian lintasan terpendek yang menghubungkan antara dua kota atau dua tempat yang berlainan. Dalam teori graf biasanya disebut *Single-source Shortest Paths Problem*.

Berikut adalah *pseudo-code* dari algoritma *dijkstra*:

Algoritma Dijkstra:

Procedure Dijkstra(G : Weighted connected simple graph, with all weight positive)

{ G has vertices $a = v_0, v_1, \dots, v_n = z$ and weights $w(v_i, v_j)$ }

where $w(v_i, v_j) = \infty$ if $\{v_i, v_j\}$ is not an edge in G

for $i := 1$ **to** n

$L(v_i) :=$

2.7. Metode Penelitian

Penggunaan metode dalam sebuah penelitian sangat diperlukan agar proses penelitian dapat berjalan sesuai dengan yang diharapkan. Fungsi metode penelitian ini adalah sebagai kerangka acuan dalam melaksanakan penelitian agar tidak menyimpang dari tujuan awal.

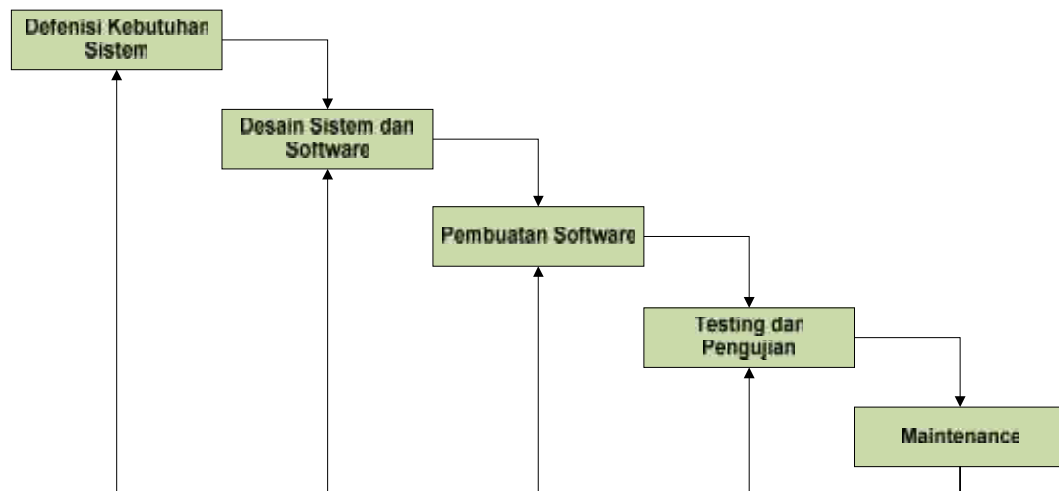
Dalam penelitian ini, akan menerapkan metode penelitian *waterfall* dengan menggunakan pendekatan metode RUP pada setiap fasenya.

2.7.1. Waterfall

Metode penelitian ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, *testing / verification*, dan *maintenance*. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan

berjalan berurutan. Sebagai contoh tahap desain harus menunggu selesainya tahap sebelumnya yaitu tahap *requirement* (Youssef Bassil, 2012).

Secara umum tahapan pada model waterfall dapat dilihat pada gambar berikut :



Gambar 2.3 Alur Proses *Waterfall*

Tahapan-tahapan pada metode penelitian waterfall adalah sebagai berikut (Youssef Bassil, 2012):

A. Definisi Kebutuhan Sistem

Pemodelan ini diawali dengan membuat suatu kebutuhan dari keseluruhan sistem yang akan diaplikasikan ke dalam bentuk *software*. Hal ini sangat penting, mengingat *software* harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database*, dan lain-lain.

B. Desain Sistem dan *Software*

Tahap ini merupakan bagian dimana seluruh kebutuhan sistem yang telah ditentukan kemudian direpresentasikan kedalam bentuk cetak biru dari *software* yang akan dibuat, desain yang dibuat harus dapat mengimplementasikan seluruh kebutuhan yang telah didefinisikan sebelumnya. Desain tersebut antara lain

desain tampilan *software*, desain struktur menu, desain *database*, dan lain sebagainya.

C. Pembuatan *Software* (*Coding*)

Pada tahap ini, pembuatan *software* mulai dilakukan dengan menggunakan bahasa pemrograman yang telah didefinisikan pada tahap awal. Pembuatan *software* dilakukan secara bertahap dan terus didokumentasikan setiap pengerjaannya agar tidak terjadi kesalahan dalam pengerjaannya. Proses pembuatan *software* ini mengacu kepada desain sistem yang telah diselesaikan sebelumnya.

D. *Testing* dan Pengujian

Pengujian *software* bertujuan untuk menguji dan mengukur kemampuan *software* dalam menjalankan tugasnya. Seluruh fungsi dan fitur dari *software* harus diuji agar bebas dari kesalahan/eror. Dan hasil dari pengujian harus benar-benar sesuai dengan kebutuhan yang telah didefinisikan sebelumnya.

E. *Maintenance*

Pemeliharaan terhadap *software* yang telah dibuat sangat perlu dilakukan sebagai bentuk dukungan terhadap kinerja *software*. Selain itu pada tahap *maintenance* juga dapat melakukan *update* terhadap *software* yang telah dibuat untuk mengatasi kesalahan atau eror yang ditemukan dikemudian hari.

2.7.2. *Rational Unified Process* (RUP)

Untuk pengembangan aplikasi *Pekanbaru Transportation Guide* di Android, metode yang digunakan adalah *Rational Unified Process* (RUP).

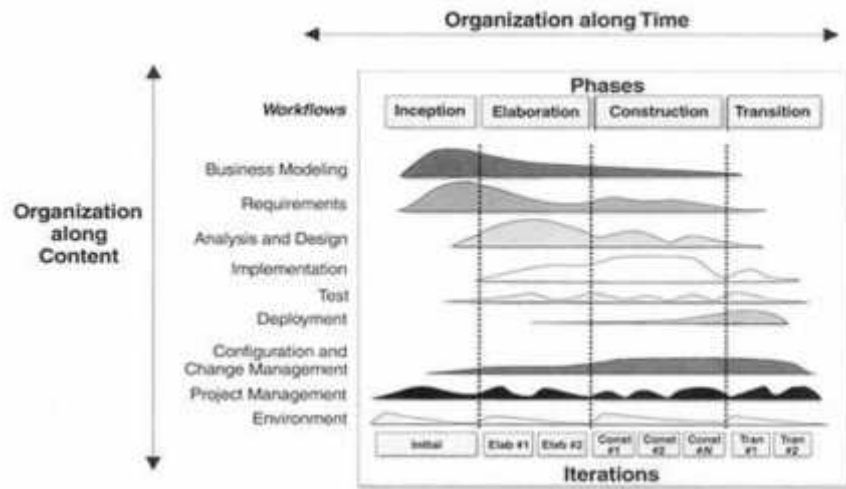
2.7.2.1. Pengertian RUP

Rational Unified Process adalah sebuah proses rekayasa perangkat lunak. RUP menyediakan pendekatan disiplin untuk memberikan tugas dan tanggung jawab dalam organisasi pengembang perangkat lunak. Tujuannya untuk menghasilkan perangkat lunak yang berkualitas dan sesuai dengan kebutuhan penggunaannya dalam anggaran dan jadwal yang dapat diprediksi dan dikendalikan (Kruchten, 2000).

RUP mengarahkan pengembangan perangkat lunak secara praktis dan efektif. Terdapat 6 *best practice* atau disebut juga *basic principle* dalam metode RUP, antara lain (Kruchten, 2000):

1. *Develop software iteratively*, bertujuan untuk mengurangi resiko pada awal proyek.
2. *Manage requirements*, bertujuan untuk mengatur kebutuhan yang diperlukan selama proyek.
3. *Use component-based architectures*, untuk membangun komponen arsitektur sebuah proyek.
4. *Visually model software*, bertujuan untuk merancang sebuah model visual perangkat lunak, untuk mendapatkan struktur dan perilaku dari aritektur perangkat lunak.
5. *Continuously verify software quality*, verifikasi dari aplikasi yg berkualitas.
6. *Control changes to software*. kemampuan untuk mengatur serta mengubah perangkat lunak saat dibutuhkan.

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Melalui gambar 2.2 dibawah dapat dilihat bahwa RUP memiliki 2 dimensi, yaitu:



Gambar 2.4 Struktur proses 2 dimensi RUP

Dimensi pertama digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari phase selanjutnya. Setiap phase dapat berdiri dari satu atau beberapa iterasi. Dimensi ini terdiri atas *Inception*, *Elaboration*, *Construction*, dan *Transition*.

Dimensi kedua digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing*, *what*, *how* dan *when*. Dimensi ini terdiri atas *Business Modeling*, *Requirement*, *Analysis and Design*, *Implementation*, *Test*, *Deployment*, *Configuration* dan *Change Manegement*, *Project Management*, *Environtment*.

2.7.2.2. Fase RUP

Fase-fase pada RUP berdasarkan waktu pengerjaan proyek dapat dibagi menjadi 4 fase, yaitu *Inception*, *Elaboration*, *Construction* dan *Transition* (Rational Team, 2001).

1. Fase *Inception*

Fase *inception* merupakan fase untuk mengidentifikasi masalah, untuk itu diperlukan juga identifikasi entitas dari luar yang berhubungan dengan sistem. Pada fase ini melibatkan semua identifikasi *use case* dan gambarannya. Selain itu juga termasuk kriteria keberhasilan proyek, perkiraan resiko, perkiraan terhadap *resource* yang dibutuhkan dan merencanakan penjadwalan *milestone*

2. Fase *Elaboration*

Tujuan dari fase *elaboration* (pengembangan) adalah menganalisa area permasalahan, mengembangkan rencana proyek, dan menghilangkan unsur-unsur yang memiliki resiko besar terhadap proyek.

3. Fase *Contruction*

Selama fase konstruksi, semua komponen dan fitur yang dikembangkan terintegrasi ke dalam produk dan secara menyeluruh semua fitur telah diuji. Di lain sisi, proses konstruksi adalah sebuah proses *manufacturing*, dimana terdapat penekanan dalam mengelola *resource* dan mengatur operasi untuk mengoptimalkan jadwal dan kualitas. Pada tahap ini pola pikir (*mindset*) mengalami perubahan dari pengembangan *intellectual property* pada fase *Inception* dan *Elaboration*, menjadi pengembangan *deployable product*.

4. Fase *Transition*

Tujuan dari fase ini adalah untuk transisi dari produk perangkat lunak ke pengguna akhir. Apabila produk telah di luncurkan kepada pengguna, maka isu-isu akan muncul dari pengguna. Nantinya isu ini akan digunakan untuk tahap perbaikan terhadap produk.

BAB III

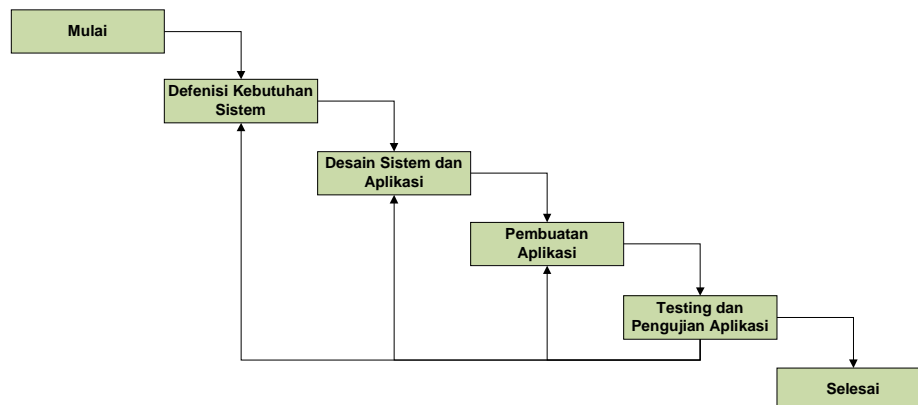
METODOLOGI PENELITIAN

3.1. Tahapan *Waterfall*

Pada bab ini akan dijelaskan mengenai tahapan-tahapan dalam pembuatan aplikasi *Pekanbaru Taksu Guide* dengan menerapkan metode *waterfall* dengan menggunakan pendekatan metode RUP pada setiap tahapannya.

3.1.1. Alur Tahapan *Waterfall*

Alur tahapan *Waterfall* yang akan digunakan dalam membuat rancang bangun aplikasi *Pekanbaru Taksu Guide* pada *Android* ditunjukkan pada Gambar 3.1 dibawah ini:



Gambar 3.1. Tahapan Penelitian

3.1.1.1. Defenisi Kebutuhan Sistem

Pada fase ini akan dilakukan tugas-tugas sebagai berikut:

- Pengenalan masalah, memahami permasalahan yang terjadi, mengapa diperlukan aplikasi *Pekanbaru Taksu Guide* pada perangkat *smartphone* dengan sistem operasi *Android*, yaitu :

- i Teknologi yang berkembang pesat dan dibutuhkan suatu sarana informasi yang bisa diandalkan kapan saja dan dimana saja.
 - ii Dibutuhkannya informasi mengenai transportasi umum taksi berupa rute perjalanan terpendek, perkiraan biaya perjalanan dan lama perjalanan.
 - iii Dibutuhkannya suatu informasi mengenai kondisi lalu lintas terkini guna mencegah pengguna agar terhindar dari kemacetan ataupun pengalihan jalan.
 - iv Belum adanya penelitian yang membahas mengenai aplikasi *taksi guide* dimana memberikan informasi mengenai angkutan umum taksi, terutama untuk wilayah Kota Pekanbaru.
- b. Pembuatan proposal, yaitu mencakup latar belakang permasalahan, rumusan masalah, batasan penelitian, tujuan, manfaat, sistematika penulisan, landasan teori, dan metodologi penelitian.
 - c. Studi literatur, mencakup penelusuran teori-teori yang berhubungan dengan permasalahan, yang bersumber dari buku, jurnal ilmiah, artikel internet dan penelitian-penelitian lainnya.
 - d. *Project Plan*, yaitu mencakup jadwal pelaksanaan tugas-tugas yang akan dijalani. Beserta tahap-tahapan sebelum membuat model *use-case*.
 - e. Pembuatan model, terbagi lagi menjadi pemodelan *use-case* dan pembuatan model sistem.

3.1.1.2. Desain Sistem dan Aplikasi

Pada fase ini akan dilakukan tugas-tugas sebagai berikut:

- a. Perancangan Aplikasi
- b. Perancangan *database* aplikasi

3.1.1.3. Pembuatan Aplikasi

Pada fase ini akan dilakukan tugas-tugas sebagai berikut:

- a. Pengkodean aplikasi yang berpedoman pada model *use-case*

menggunakan bahasa pemrograman Java.

- b. Melakukan pengujian terhadap kesalahan-kesalahan yang mungkin terjadi selama proses pengkodean.

3.1.1.4. Testing dan Pengujian Aplikasi

- a. Melakukan pengujian terhadap aplikasi yang dibuat.
- b. Menyimpulkan fitur-fitur tambahan guna pengembangan dari aplikasi Pekanbaru Taksi Guide untuk kedepannya.

BAB IV

ANALISA DAN PERANCANGAN

Bab ini merupakan bagian dari Tahapan Defenisi Kebutuhan Sistem dan Desain Sistem dan Aplikasi, dimana akan dilakukan analisa sejalan dengan pembuatan deskripsi arsitektur yang dibutuhkan aplikasi dan pembuatan UML (*Unified Modelling Language*) sebagai bagian dari tahapan defenisi kebutuhan sistem, dan dilakukan perancangan struktur menu aplikasi yang akan dibangun dan perancangan *prototype* antarmuka aplikasi yang akan dibangun sebagai bagian dari tahapan desain sistem dan aplikasi.

4.1. Analisa

Aplikasi Pekanbaru Taksi Guide ini akan berjalan diatas sistem operasi *Android* dengan menggunakan teknologi GPS sebagai penentu lokasi awal pengguna dan dengan menggunakan metode *client-server*, yang seluruh data koordinat lokasi tujuan, koordinat jalan berada di *server* dan proses perhitungan algoritma *Dijkstra* dilakukan oleh *server*. Hasil dari perhitungan akan dikirimkan ke perangkat pengguna dan ditampilkan berupa penunjuk arah (*direction*) rute terpendek menuju lokasi tujuan.

Aplikasi ini berbasis GIS *mobile* yang menerapkan peta dari *Google* (*Google Maps*) sebagai penunjuk lokasi koordinat. Berupa koordinat awal pengguna, koordinat lokasi tujuan maupun koordinat jalan. Seluruh data koordinat disesuaikan dengan koordinat di *Google Maps*.

Aplikasi ini ditujukan untuk calon penumpang yang ingin menggunakan angkutan umum taksi. Melalui aplikasi ini, calon penumpang dapat memperoleh informasi seperti rute terpendek menuju lokasi tujuan, informasi perkiraan biaya perjalanan, informasi perkiraan waktu perjalanan, informasi lalu-lintas terkini mengenai perkiraan titik kemacetan jalan, sehingga aplikasi mampu mencari rute

terpendek tanpa harus melewati titik kemacetan jalan, serta informasi nomor kontak taksi yang ada di Kota Pekanbaru.

Unuk penentuan besarnya tarif, aplikasi ini merujuk pada SK Walikota Pekanbaru Nomor 31/2009 tanggal 24 Juli 2009 tentang pemberlakuan tarif dasar taksi yang baru yang ditunjukkan pada tabel 4.1 dibawah ini:

Tabel 4.1 Tarif Dasar Taksi Kota Pekanbaru

Jenis Tarif Dasar	Biaya (Rp)
Tarif buka pintu	6.000
Tarif perkilometer	3.500
Tarif minimal argo	20.000
Tarif pembatalan sepihak	10.000
Tarif tunggu per jam	20.000

Untuk data koordinat lokasi tujuan, dibagi menjadi beberapa kategori seperti yang telah dijelaskan di BAB I, yaitu kategori hotel, kategori rumah sakit, kategori kuliner, kategori tempat rekreasi, kategori mall, dan kategori *public transport*.

Untuk data koordinat jalan yang akan digunakan, dilakukan digitasi peta berdasarkan *Google Maps*. Sehingga didapatkan data koordinat jalan yang sesuai dengan kebutuhan aplikasi dan setiap kondisi jalan yang ada telah sesuai dengan peraturan lalu-lintas yang berlaku. Untuk data koordinat jalan ini, hanya dibatasi pada jalan-jalan protokol yang ada di Kota Pekanbaru.

4.1.1. Defenisi Kebutuhan Sistem

Tahap ini memiliki beberapa tahapan yang akan dilaksanakan dalam pembuatan sistem, antara lain pengenalan masalah, pembuatan proposal, studi literatur, *project plan* dan pembuatan model.

Untuk tahapan pengenalan masalah, pembuatan proposal dan studi literatur sudah dibahas sebelumnya pada bab 1, bab 2, dan bab 3.

4.1.1.1. Project Plan

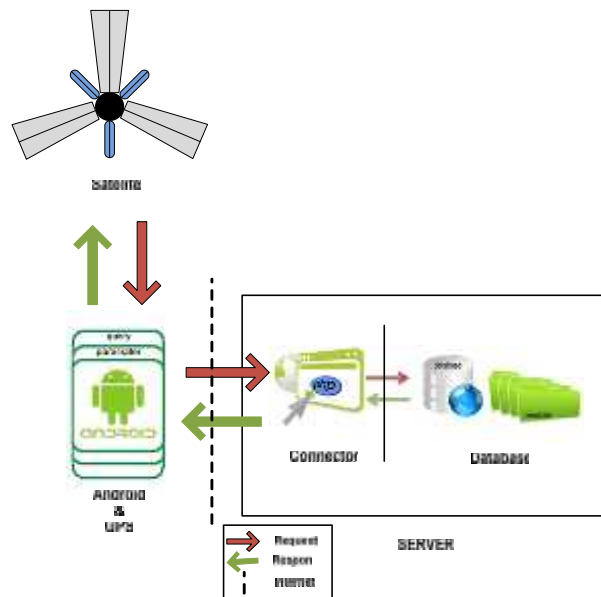
Pada *project plan* ini, akan dijelaskan mengenai gambaran umum sistem, deskripsi kebutuhan sistem, dan fungsi sistem yang nantinya akan dibuat.

A. Gambaran Umum Sistem

Sistem yang dibangun merupakan sebuah aplikasi *mobile* yang berbasis *client-server* yang menerapkan teknologi *Location Based Service* (LBS). Aplikasi ini dapat digunakan untuk melakukan aktifitas pencarian lokasi. Pencarian lokasi yang dimaksud adalah mencari rute terpendek, mencari lokasi seperti lokasi hotel, rekreasi, rumah sakit, *mall* dan kuliner, menemukan rute navigasi menuju lokasi tujuan, melihat perkiraan biaya perjalanan menggunakan taksi serta jarak dan lamanya waktu perjalanan. Sistem aplikasi ini diberi nama Pekanbaru Taksi Guide (PTG), sehingga untuk selanjutnya penyebutan sistem aplikasi ini *mobile* ini adalah PTG.

Secara umum, PTG menerapkan konsep *client-server*. Dimana aplikasi akan terinstall di perangkat pengguna, kemudian dibutuhkan sebuah *server* untuk menerima, mengolah, serta mengirimkan *request* dari dan perangkat pengguna. Seluruh aktifitas berpusat di-*server*. Seperti menerima dan mengirimkan *request*, pengolahan data, dan penyimpanan *database*.

PTG juga menerapkan teknologi berbasis lokasi sehingga memudahkan pengguna untuk melihat dimana lokasinya berada dan mampu untuk melihat lokasi-lokasi yang berada disekitarnya. Melalui teknologi ini, proses navigasi menjadi lebih akurat dalam mencari rute terpendek untuk menuju lokasi yang dituju. Secara sederhana deskripsi umum sistem dapat dilihat pada gambar 4.1 dibawah ini.



Gambar 4.1 Arsitektur Sistem

Dari gambar diatas, terdapat 3 bagian inti dari proses kerja aplikasi. Diantaranya:

1. Perangkat *Android*

Merupakan tempat berjalannya aplikasi PTG. Dari perangkat inilah pengguna dapat berinteraksi dengan didukung oleh koneksi internet untuk dapat mengakses *server* melalui PHP serta dukungan perangkat GPS pada *android* untuk menentukan lokasinya.

2. *Server*, terdiri dari 2 bagian, yaitu:

- a. PHP sebagai *connector*, berfungsi sebagai jembatan penghubung antara sistem yang berjalan pada perangkat *android* dan *database server*. *Conenector* ini sebagai perantara yang mengirimkan *request* dan respon antara *client* di *android* dan *server database*.
- b. *Database*, berfungsi sebagai tempat penyimpanan data atau *database* aplikasi PTG. *Database* adalah bagian yang bertanggung jawab memberikan menerima *request* dan memberikan respon balik terhadap *request* yang diminta. *Database* yang digunakan disini adalah MySQL.

3. Penentu lokasi, yang terdiri dari 2 bagian, yaitu:

- a. GPS, merupakan perangkat yang ditanamkan ke *smartphone android* untuk bisa mengirimkan *request* data berupa lokasi dan posisi pengguna berdasarkan koordinat *lattitude* dan *longitude*.

- b. Satelit, merupakan perangkat yang memetakan serta memberikan respon berdasarkan permintaan GPS pada perangkat *smartphone android*.

Aplikasi ini tidak memiliki hak akses khusus untuk pengguna, sehingga pengguna mendapatkan hak akses penuh terhadap seluruh fitur pada aplikasi.

B. Deskripsi Kebutuhan Sistem

Untuk membangun sebuah sistem yang efisien, kebutuhan sistem merupakan hal yang harus diperhatikan. Mengetahui kebutuhan sistem akan membantu dalam pembangunan sistem.

1. Sistem yang akan dibangun

Untuk kebutuhan sistem yang akan dibangun terdapat dua bagian yakni dari kebutuhan sistem dari perangkat *Android* dan kebutuhan sistem dari sisi *server database*.

a. Sistem pada perangkat *Android*

1. Bahasa pemrograman yang digunakan adalah bahasa *Java*.
2. Selain melakukan fungsi navigasi dan menunjukkan rute terpendek, aplikasi ini memiliki fitur diantaranya menu pencarian, melihat peta, nomor kontak taksi dan tentang aplikasi.
3. Pada saat aplikasi dibuka, halaman pertama yang muncul adalah halaman home dengan menu yang tersedia adalah Nomor Kontak Taksi. Kemudian menu yang lain dapat diakses dengan menekan tombol menu pada *smartphone androidnya*. menu tersebut adalah Navigasi, Lihat Peta, Pencarian, Tentang Apps, dan Keluar.
4. Menu Navigasi berfungsi mencari rute terpendek dari posisi awal pengguna yang menangkap posisi GPS pengguna kemudian menuju lokasi yang ditentukannya dengan memilih lokasi tujuan dari *database server*.

5. Pada menu Navigasi, terdapat tampilan Kategori Tujuan yang membantu pengguna untuk memudahkan dalam pencarian lokasi tujuan berdasarkan kategori pada *server database*.
6. Pada menu map, akan menampilkan lokasi-lokasi dari yang telah ditandai yang muncul diatas peta.
7. Menu Pencarian berfungsi untuk mencari lokasi tujuan yang terdapat di *database* berdasarkan kategori, seperti hotel, rumah sakit, kuliner, mall, rekreasi.
8. Menu Tentang Apps merupakan penjelasan singkat tentang aplikasi itu sendiri, seperti versi aplikasi, dan lain sebagainya.
9. Aplikasi mampu memberitahukan pengguna apabila dalam proses navigasi, posisi pengguna tidak berada pada rute yang telah ditetapkan sebelumnya.

b. *Connector*

Bagian ini berfungsi sebagai penghubung atau perantara antara aplikasi yang berjalan dengan *server database*. Penghubung ini dibangun menggunakan bahasa pemrograman PHP. Berikut beberapa *connector* yang akan dibangun:

1. *Connector dijkstra*, menangani *request* untuk pencarian rute terpendek yang dikirimkan oleh pengguna, kemudian melakukan pencarian rute terpendek berdasarkan data yang ada di *server database*, kemudian mengirimkan respon dari *database* ke aplikasi.
2. *Connector kategori*, menangani *request* untuk menampilkan lokasi tujuan berdasarkan kategori yang terdapat di *server database*.
3. *Connector koordinat_lokasi*, bagian ini menangani *request* dari pengguna untuk menampilkan seluruh lokasi tujuan yang ada di *server database* ke menu Lihat Peta, sehingga pengguna bisa melihat seluruh *marker* lokasi yang ada di *server database* di peta pada perangkat *android* pengguna.
4. *Connector search*, bagian ini menangani pencarian yang dilakukan lokasi tujuan tanpa dibatasi kategori tujuan oleh pengguna.

2. Analisa Fungsional

Analisa fungsional merupakan penjelasan mengenai fitur-fitur yang akan ada pada aplikasi *Pekanbaru Taksi Guide*. Fitur-fitur tersebut antara lain:

1. Proses navigasi akan dilakukan melalui titik awal pengguna yang ditangkap oleh GPS yang aktif pada perangkat *android*, kemudian pengguna dapat memilih satu lokasi tujuan dan aplikasi akan mengarahkan rute terpendek untuk menuju kesana.
2. Pengguna dapat melakukan pencarian berdasarkan kategori lokasi. Seperti kategori mall, kuliner, rekreasi, rumah sakit, dan hotel.
3. Pengguna dapat melihat semua lokasi tujuan yang tertera pada menu lihat peta yang akan ditampilkan di peta berupa *marker* atau tanda dari lokasi, dan setiap kategori memiliki *marker* yang berbeda.
4. Pengguna dapat men-*tracking* posisi ketika sedang berjalan menuju tempat tujuan. Sehingga pengguna bisa memastikan bahwa ia sedang berada di rute yang benar.
5. Ketika posisi pengguna keluar dari rute yang telah ditentukan sebelumnya, maka aplikasi akan memberikan pemberitahuan (*alert*) kepada pengguna bahwasannya posisi pengguna sedang tidak berada pada rute perjalanan.

3. Performansi Aplikasi

Aplikasi *Pekanbaru Taksi Guide ini* merupakan aplikasi yang berjalan di lingkungan perangkat bersistem operasi *Android*. Terdapat beberapa keterbatasan yang ditemui pada perangkat ini, sehingga perlu diperhatikan untuk menjadi acuan dalam pengembangan aplikasi ini, yaitu diantaranya:

1. Sumber daya yang terbatas, hingga saat ini perangkat *Android* yang banyak beredar memiliki kapasitas memori terbatas.
2. Sumber daya baterai yang secara efektif hanya mampu bertahan selama kurang lebih 200 jam dalam keadaan *standby*. Dan tidak lebih dari 14 jam ketika digunakan terus menerus.
3. Tampilan antarmuka dengan pengguna sangat berpengaruh terhadap waktu tunggu aplikasi hingga aplikasi benar-benar siap digunakan,

semakin banyak komponen yang digunakan akan semakin lama pula waktu tunggu yang dibutuhkan.

4. Penggunaan sistem GPS juga akan berdampak pada cepat berkurangnya sumber daya baterai.
5. Penggunaan koneksi internet yang selalu mengirimkan data ke server akan berdampak berkurangnya daya tahan baterai serta memerlukan kuota internet yang tidak sedikit.

Dari keterbatasan-keterbatasan pada perangkat *Android*, maka diusulkan beberapa alternatif untuk meningkatkan performa aplikasi terhadap keterbatasan yang ada, diantaranya:

1. Merancang aplikasi yang menggunakan memori seefektif mungkin, sehingga tidak mengganggu siklus operasi *Android* dan aplikasi lain.
2. Merancang aplikasi dengan pemanfaatan sumber daya seefisien mungkin namun tidak mengurangi fungsi dan performa aplikasi.
3. Merancang aplikasi dengan antarmuka yang sederhana .
4. Merancang aplikasi dengan penggunaan kuota internet sekecil mungkin.

C. Fungsi Sistem

Secara umum fungsi sistem ada dua bagian yaitu sistem yang akan dibangun dari sisi perangkat *Android* dan dari *server database*.

1. Fungsi Sistem dari Sisi Perangkat *Android*

Sistem yang akan dibangun dari sisi perangkat *android* memiliki fungsi-fungsi sebagai berikut:

- a) Mampu menavigasikan rute terpendek dari posisi awal pengguna menuju lokasi tujuan.
- b) Menampilkan tempat-tempat yang telah ditandai dipeta berdasarkan kategori yang telah di batasi di batasan penelitian.
- c) Mampu melakukan pencarian lokasi tujuan yang terdapat di *server database*.
- d) Mampu menampilkan rute terpendek yang telah ditandai di peta.

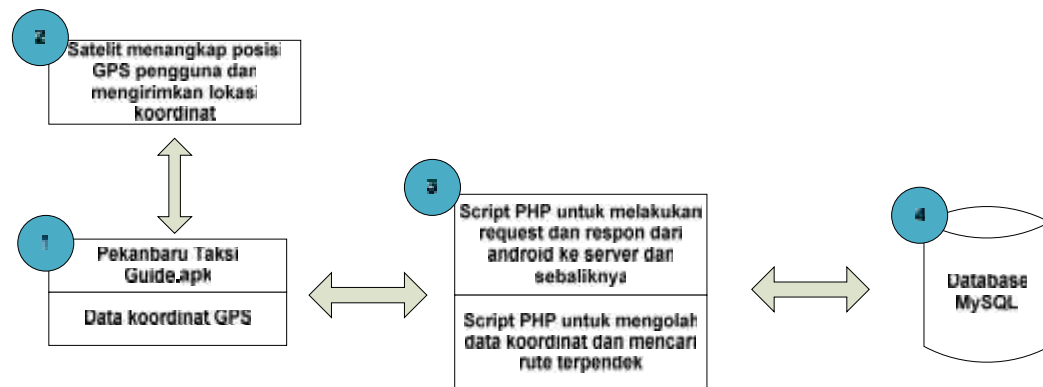
2. Fungsi Mesin Penghubung

Mesin penghubung berfungsi untuk menerima *request* dari perangkat *android* pengguna dan meneruskannya ke *server database* dan kemudian mengolahnya serta mengirimkan lagi hasilnya ke perangkat *android*.

4.1.1.2. Pembuatan Model

Pada bagian pembuatan model ini, terbagi 3 tahapan pengerjaan. Yaitu pembuatan model sistem, penjelasan deskripsi pengguna dan pemodelan UML.

A. Model Sistem



Gambar 4.2. Model Sistem

Objek-objek yang berperan pada model sistem diatas yaitu:

- Perangkat *android* sebagai media untuk mengakses *server* melalui aplikasi PTG yang telah terinstall.
- GPS sebagai penentu posisi pengguna aplikasi yang terhubung ke saelit untuk meminta request posisi pengguna berdasarkan *longitude* dan *lattitude*, yang nantinya akan digunakan untuk diproses sebagai titik awal dalam memulai navigasi.
- Script* PHP berperan sebagai media untuk *request* dan respon dari perangkat *android* ke *database* dan sebaliknya serta untuk memproses pencarian rute terpendek dengan menggunakan data yang ada di *database*.

- d. *Database* MySQL berperan sebagai tempat penyimpanan data yang dibutuhkan oleh aplikasi serta oleh script php untuk memproses pencarian rute terpendeknya.

B. Deskripsi Pengguna

Pengguna dari sistem ini adalah calon penumpang taksi, dan penumpang diberikan semua hak akses penuh terhadap seluruh fitur dan fungsi yang ada pada aplikasi ini. Sedangkan admin bertugas menyediakan *update* peta dengan *update* kondisi lalu lintas terakhir kondisi terakhir dari kota pekanbaru. Untuk lebih jelasnya dapat di lihat di tabel 4.2. di bawah ini.

Tabel 4.2. Deskripsi pengguna

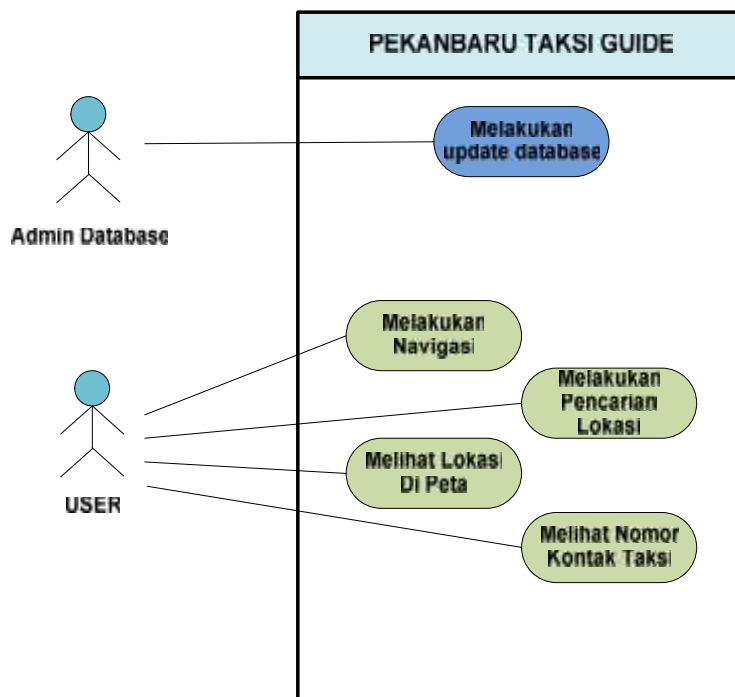
No	Kategori Pengguna	Hak Akses	Keterangan
1	Calon Penumpang (Perangkat <i>Android</i>)	a. Melakukan navigasi ke tempat tujuan yang disertai dengan informasi perkiraan biaya perjalanan dan lamanya perjalanan yang diawali dari posisi pengguna saat menggunakan aplikasi b. Melakukan pencarian lokasi tujuan c. Melihat lokasi-lokasi tujuan yang tertera di peta d. Mampu untuk men- <i>tracking</i> posisi ketika sedang menuju lokasi tujuan	Hak akses penuh
2	Admin	a. Melakukan update <i>database</i> , baik <i>database</i> koordinat jalan maupun koordinat lokasi dan info lalu lintas terkini	Hak akses penuh

C. Pemodelan UML (*Unified Modeling Language*)

Analisa perancangan pada sistem ini menggunakan UML, yaitu terdiri dari *usecase diagram*, *class diagram*, *activity diagram*, *sequence diagram* dan *deployment diagram*. Untuk lebih jelasnya dapat dilihat pada lampiran A.

1. Usecase Diagram

Usecase diagram merupakan suatu aktivitas yang menggambarkan urutan interaksi antar satu atau lebih aktor dan sistem. *Usecase* yang akan dirancang yaitu *usecase diagram* untuk pengaksesan melalui perangkat *Android*. Gambar 4.2 dibawah ini menjelaskan aliran *usecase diagram* pengaksesan melalui perangkat *Android*.



Gambar 4.3. Aliran *Usecase Diagram* (pengaksesan melalui perangkat *Android*)

Dari gambar diatas, dapat dilihat sistem ini terdiri dari 2 aktor dan 5 *usecase*. Untuk lebih jelasnya, Spesifikasi dari *usecase diagram* (pengaksesan melalui perangkat *Android*) dapat dilihat pada tabel 4.3. dibawah ini.

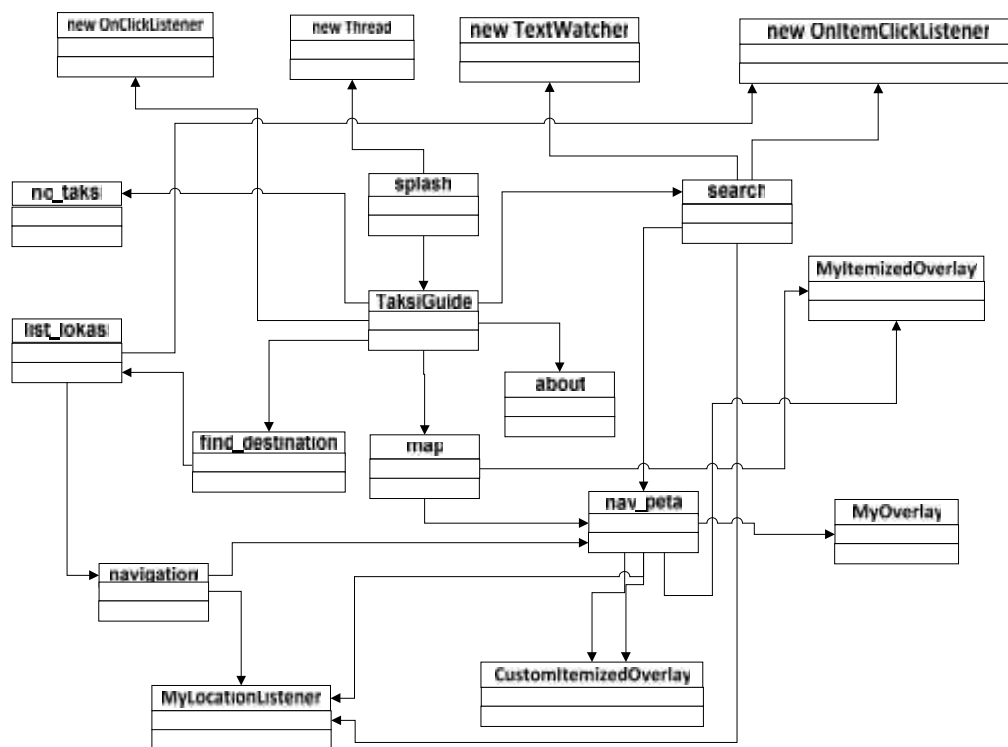
Tabel 4.3. Spesifikasi *Usecase diagram*

No.	Aktor	Nama Usecase	Deskripsi
1.	Admin Database	Login	Proses autentifikasi untuk masuk ke <i>database server</i>
		Melakukan <i>update database</i>	Menambah lokasi tujuan dan <i>update database</i> koordinat jalan

2.	User (Penumpang)	Melakukan navigasi	Proses melakukan navigasi dan menuju ke lokasi tujuan
		Melakukan Pencarian	Proses untuk melakukan pencarian lokasi tujuan
		Melihat Lokasi di Peta	Proses untuk menavigasi peta dan melihat lokasi-lokasi yang ditandai di peta
		Melihat nomor kontak taksi	Proses menampilkan nomor kontak taksi yang ada di pekanbaru

2. Class Diagram

Class Diagram menggambarkan struktur dan deskripsi *class*, *package*, dan objek yang saling terhubung. *Class diagram* yang dijelaskan pada analisa ini adalah *class diagram* sistem yang terpasang pada perangkat *Android*. Gambar 4.4 dibawah ini menjelaskan *class diagram* sistem yang terpasang pada perangkat *Android*.



Gambar 4.4. Class diagram sistem di perangkat *Android*

Tabel 4.4. Detail *Class Diagram*

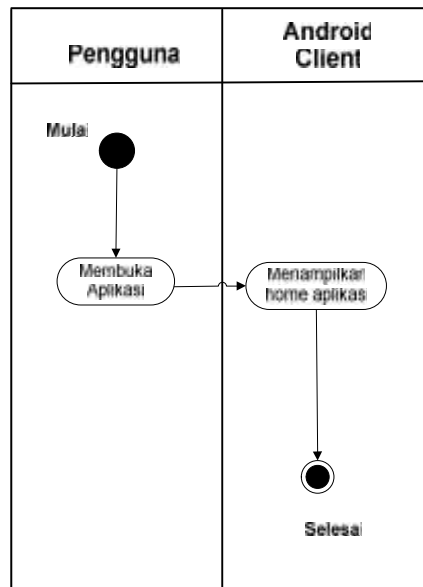
No	Class	Atribut	Method
1.	TaksiGuide	MENU_NAVIGATION : int MENU_MAP : int MENU_SEARCH : int MENU_ABOUT : int MENU_EXIT : int taksi_number : Button	onCreate(Bundle) no_taksi() onCreateOptionsMenu(Menu) onOptionsItemSelected(MenuItem)
2.	Splash	player : MediaPlayer	onCreate(Bundle) : void onPuase() : void onResume() : void playSound(int) : void startAnimasi() : void
3.	search	longitude : double latitude : double btn_findplace1 : Button btn_home : Button list_search : ListView edit_search : EditText jObject : JSONObject url : String xResult : String data : int nama_tempat : String[] titik_lokasi : String[] kategori : String[] key : String longi : String lati : String lm : LocationManager locationListener : LocationListener lon : String lat : String fid : String kategori2 : String	onCreate(Bundle) list(String[]) map(String, String, String, String) request(HttpResponse) getRequest(String)
4.	no_taksi		onCreate(Bundle)
5.	navigation	btn_find_location : Button btn_find_destination : Button btn_go : Button longitude : double latitude : double lm : LocationManager locationListener : LocationListener longi : TextView lati : TextView lon : String lat : String fid : String kategori : String kategori2 : String	onLocationChanged(Location) onProviderDisabled(String) onProviderEnabled(String) onStatusChanged(String, int, Bundle) onCreate(Bundle) onClick(View)

		des : EditText titik_akhir : String deskrip : String onCreate(Bundle) onClick(View)	
6.	nav_peta	btn_home : Button projection : Projection mc : MapController mapView : MapView gP : GeoPoint url : String jObject : JSONObject myoverlay : MyOverlay titik_awal : double[][] titik_akhir : double[][] data : int xResult : String longitude : double latitude : double longitude_akhir : double latitude_akhir : double lm : LocationManager locationListener : LocationListener drawable : Drawable drawable2 : Drawable kategori : String x : String y : String fid : String panjang : String mapOverlays : List<Overlay> waktu : TextView biaya : TextView biaya2 : int waktu2 : float	onCreate(Bundle) onClick(View) getRequest(String) request(HttpResponse)
7.	map	url : String jObject : JSONObject titik : double[][] keterangan : String[] kategori : String[] titik_akhir : String[] data : int xResult : String lokasi2 : String titik_akhir2 : String kategori2 : String	onCreate(Bundle) isRouteDisplayed() direc_lokasi(String, String, String) getRequest(String) request(HttpResponse)
8.	list_lokasi	kategori2 : String nama_pilihan : TextView lokasi : ListView jObject : JSONObject xResult : String url : String nama_lokasi : String[] titik_akhir : String[]	onCreate(Bundle) direc_lokasi(String, String, String) getRequest(String) request(HttpResponse)

		lokasi2 : String titik_akhir2 : String kategori : String	
9.	find_destination	kategori : String btn_destination : int[] destination : Button[]	onCreate(Bundle) onClick(View) list(String)
10.	About		onCreate(Bundle)
11.	new OnClickListener		onClick(View): void
12.	new Thread	wait : int	run():void
13.	new TextWatcher		afterTextChanged(Editable) beforeTextChanged(CharSequence, int, int, int) onTextChanged(CharSequence, int, int, int)
14.	new OnItemClickListener		onItemClick(AdapterView<?>, View, int, long):void
15.	MyLocationListe ner		onLocationChanged(Location) onProviderDisabled(String) onProviderEnabled(String) onStatusChanged(String, int, Bundle)
16.	MyItemizedOverl ay	mOverlays: List<OverlayItem>	MyItemizedOverlay(Drawable) addOverlayItem(int, int, String, Drawable) addOverlayItem(OverlayItem) addOverlayItem(OverlayItem, Drawable) size() createItem(int) onTap(int)
17.	MyOverlay		MyOverlay() draw(Canvas, MapView, boolean)
18.	CustomItemized Overlay	mOverlays: ArrayList<OverlayItem> lihat_peta : MapActivity	CustomItemizedOverlay(Drawable) CustomItemizedOverlay(Drawable, MapActivity) addOverlay(OverlayItem) createItem(int) size() onTap(int)

3. Activity Diagram

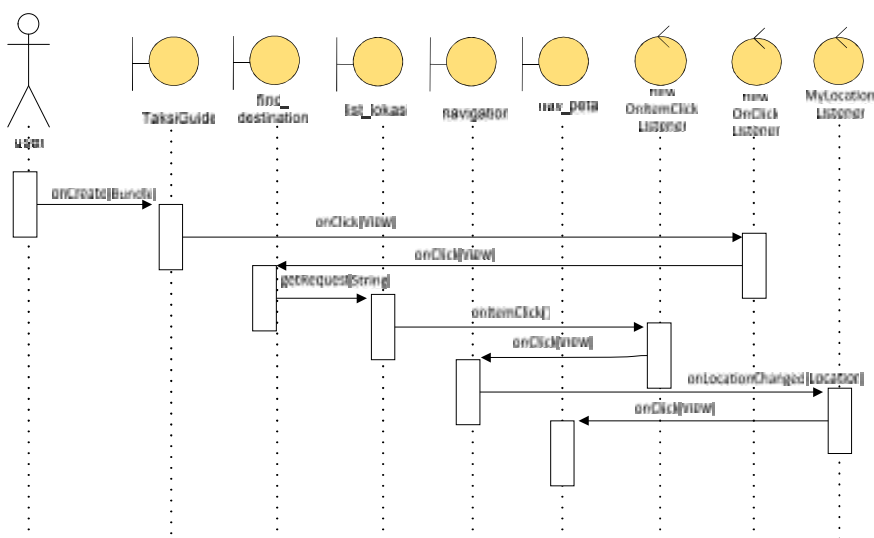
Activity diagram merupakan alur kerja pada setiap *usecase*. *Activity diagram* pada analisa ini mencakup *activity diagram* setiap *usecase*. Gambar 4.5. dibawah ini menjelaskan *activity* menampilkan *Home* dari aplikasi pada perangkat *Android*. Untuk *activity diagram* lainnya dapat di lihat pada lampiran A.



Gambar 4.5. Activity Diagram Home Aplikasi

4. Sequence Diagram

Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display* dan sebagainya) berupa pesan yang digambarkan terhadap waktu. Gambar 4.6 dibawah ini menggambarkan *sequence diagram login* pada *server*. Untuk *Sequence diagram* lainnya dapat di lihat pada lampiran A.

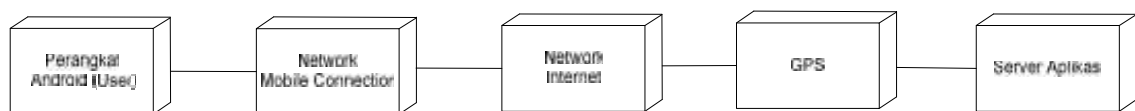


Gambar 4.6. Sequence diagram untuk mencari dan menuju lokasi tujuan

5. Deployment Diagram

Deploy diagram menggambarkan detail bagaimana komponen di *deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa).

Aplikasi diterapkan pada perangkat *android* yang menggunakan koneksi internet untuk bisa mengakses *server* aplikasi dan perangkat GPS untuk menentukan lokasi pengguna sehingga *deploy diagram* yang dihasilkan dapat dilihat pada Gambar 4.7 dibawah ini.



Gambar 4.7. Deployment Diagram

4.2. Perancangan

Tahap perancangan merupakan tahap dimana perancangan aplikasi mulai dilakukan. Perancangan tersebut antara lain perancangan struktur menu, perancangan *graphic user interface* aplikasi di perangkat *android*, perancangan *graphic user interface* untuk admin *database*, dan perancangan *database* aplikasi.

Perancangan *interface* aplikasi di *android* ditujukan kepada pengguna aplikasi, sedangkan *interface* admin *database* ditujukan untuk admin *database* yang akan melakukan *update* kondisi lalu-lintas, serta *update* koordinat lokasi tujuan dan koordinat jalan.

4.2.1. Desain Sistem dan Aplikasi

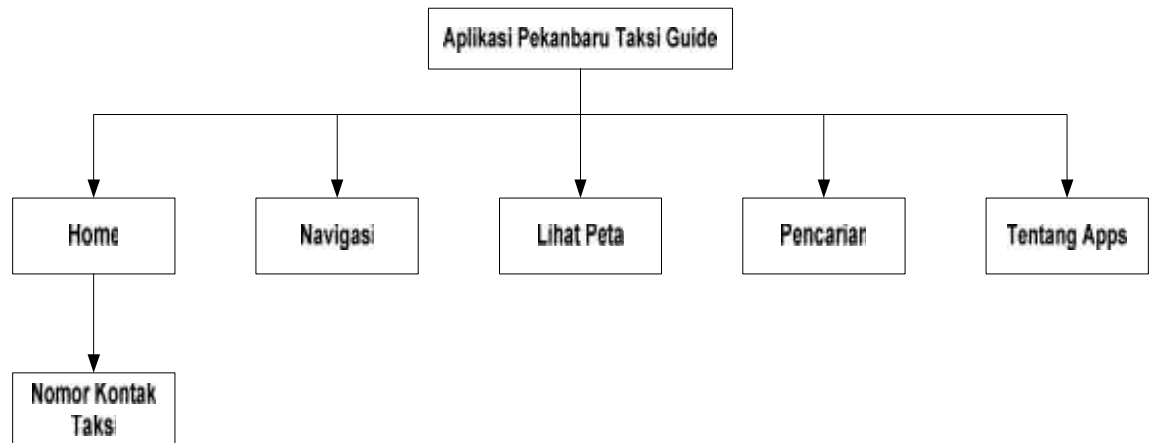
Tahapan pada *fase* ini terbagi menjadi 2 yaitu perancangan aplikasi dan perancangan *database* aplikasi.

4.2.1.1. Perancangan Aplikasi

Pada tahap perancangan aplikasi, terbagi menjadi 2 tahap. Yaitu perancangan struktur menu dan perancangan antarmuka pengguna aplikasi.

A. Perancangan Struktur Menu Sistem

Rancangan struktur menu merupakan tahapan untuk merancang bagaimana struktur menu yang akan dibangun. berikut struktur menu dari sistem yang akan dibangun yang dapat dilihat pada gambar di bawah ini.

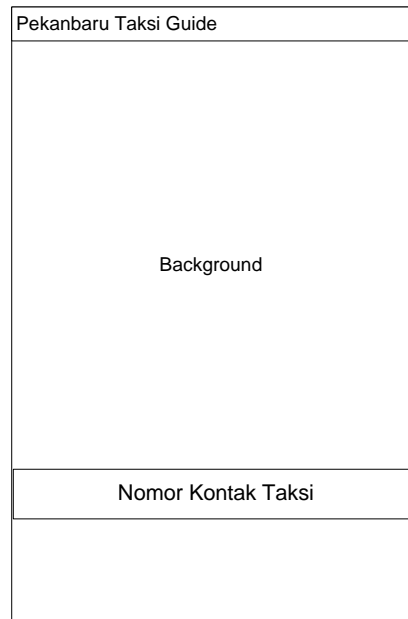


Gambar 4.8. Rancangan Struktur Menu

B. Perancangan Antarmuka Pengguna Aplikasi

Rancangan antarmuka pengguna aplikasi berfungsi sebagai landasan awal dalam merancang tampilan antarmuka sistem. Secara umum, *background* aplikasi yang akan digunakan nanti adalah berwarna kuning dimana kuning sangat identik dengan Kota Pekanbaru yang merupakan ibu kota Propinsi Riau dan sebagai sebuah Negeri Melayu.

Pada analisa dan perancangan ini, antarmuka untuk pengguna sistem pada perangkat *android* antara lain, antarmuka Home, antarmuka Navigasi, antarmuka Peta, antarmuka Navigasi Peta, antarmuka Pencarian, antarmuka Kategori Tujuan, antarmuka Nomor Kontak Taksi, dan antarmuka Tentang Apps. Untuk lebih lengkapnya dapat dilihat pada lampiran B.



Gambar 4.9. Perancangan antarmuka halaman *Home* Aplikasi

Deskripsi gambar diatas tentang perancangan antarmuka *Home* pada perangkat *Android* dapat dilihat pada tabel 4.5 di bawah ini.

Tabel 4.5. Deskripsi antarmuka *home* pada perangkat *Android*

No	Nama Item	Deskripsi
1.	Pekanbaru Taksi <i>Guide</i>	Merupakan sebuah <i>widget</i> berupa <i>Textview</i>
2.	<i>Background</i>	Merupakan sebuah <i>widget</i> berupa <i>AbsoluteLayout</i>
3.	Nomor Kontak Taksi	Merupakan sebuah <i>widget</i> berupa <i>Button</i>

1. Header, menggunakan *font* dengan warna #000000 atau hitam, dengan posisi tulisan diratakan pada *margin* sebelah kiri, dan berjarak 20px dari tepi *layout*.
2. *Background*, menggunakan *image* yaitu *drawable/home2.jpg*
3. Nomor Kontak Taksi , merupakan sebuah *button* yang bernama *taksi_number* dengan *gravity center_vertical/center_horizontal* dan *text style bold*.

4.2.1.2. Perancangan *Database*

Pada tahap perancangan *database* aplikasi ini, proses tahapan pengerjaan dibagi menjadi 2, yaitu perancangan *database* aplikasi, dan perancangan antarmuka *database* aplikasi.

A. Perancangan Database Aplikasi

Database merupakan bagian penting dari aplikasi PTG. Seluruh fitur pada aplikasi menggunakan *database* untuk melakukan fungsinya. perancangan *database* pada aplikasi PTG hanya meliputi perancangan *database server*.

Database server adalah *database* yang ada di komputer *server*, dan untuk mengaksesnya memerlukan suatu koneksi internet pada perangkat *android*. Pada gambar 4.9 berikut, dijelaskan mengenai perancangan *database* dari aplikasi PTG, dimana menggunakan 3 tabel, yaitu tabel *account*, tabel *berita*, dan tabel *koordinat*.

Tabel 4.6. Tabel Perancangan *Database* Aplikasi PTG

account		berita		koordinat	
PK	username	PK	no	PK	no
	password		id_berita		nama_jalan
					titik_awal
					titik_akhir
					x1
					y1
					x2
					y2
					panjang
					keterangan
					kategori

Tabel 4.7. Keterangan Atribut dari tabel Koordinat

No	Nama Field	Tipe Data	Null	Keterangan
1	No (PK)	Int(10)	Not Null	Id dari graf berarah
2	nama_jalan	Varchar(1000)	Not Null	Nama jalan
3	titik_awal	Int(100)	Not Null	Titik awal graf berarah
4	titik_akhir	Int(100)	Not Null	Titik akhir dari graf berarah
5	x1	Varchar(1000)	Not Null	Koordinat Longitude titik awal graf berarah
6	y1	Varchar(1000)	Not Null	Koordinat Lattitude titik awal graff berarah
7	x2	Varchar(1000)	Not Null	Koordinat Longitude titik akhir graf berarah
8	y2	Varchar(1000)	Not Null	Koordinat Lattitude titik akhir graf berarah
9	panjang	Varchar(1000)	Not Null	Panjang antara titik awal dan

				titik akhir dari graf berarah
10	keterangan	Varchar(1000)	Null	Keterangan dari titik akhir graf berarah yg berupa nama jalan atau nama simpang
11	kategori	Varchar(50)	Null	Nama Kategori

Tabel 4.8. Keterangan Atribut dari Tabel Account

No	Nama Field	Type Data	Null	Keterangan
1	username (PK)	Varchar(1000)	Not Null	Merupakan username yang digunakan untuk login mengakses antarmuka <i>database</i> aplikasi
2	password	Varchar(1000)	Not Null	Kata kunci untuk tiap username yang digunakan

Tabel 4.9. Keterangan Atribut dari Tabel Berita

No	Nama Field	Type Data	Null	Keterangan
1	no (PK)	Varchar(10)	Not Null	Merupakan nomor urut setiap informasi lalu lintas
2	id_berita	Varchar(1000)	Not Null	Berisi tentang informasi lalu lintas saat ini mengenai kemacetan panjang, pengalihan jalan dan penutupan jalan.

B. Perancangan Antarmuka *Database* Aplikasi

Rancangan antarmuka *database* aplikasi merupakan bagian yang dikhususkan untuk admin *database* yang berfungsi sebagai antarmuka untuk melakukan perubahan pada *database* seperti penambahan data, penghapusan data, perubahan data, serta mampu melakukan pencarian data.

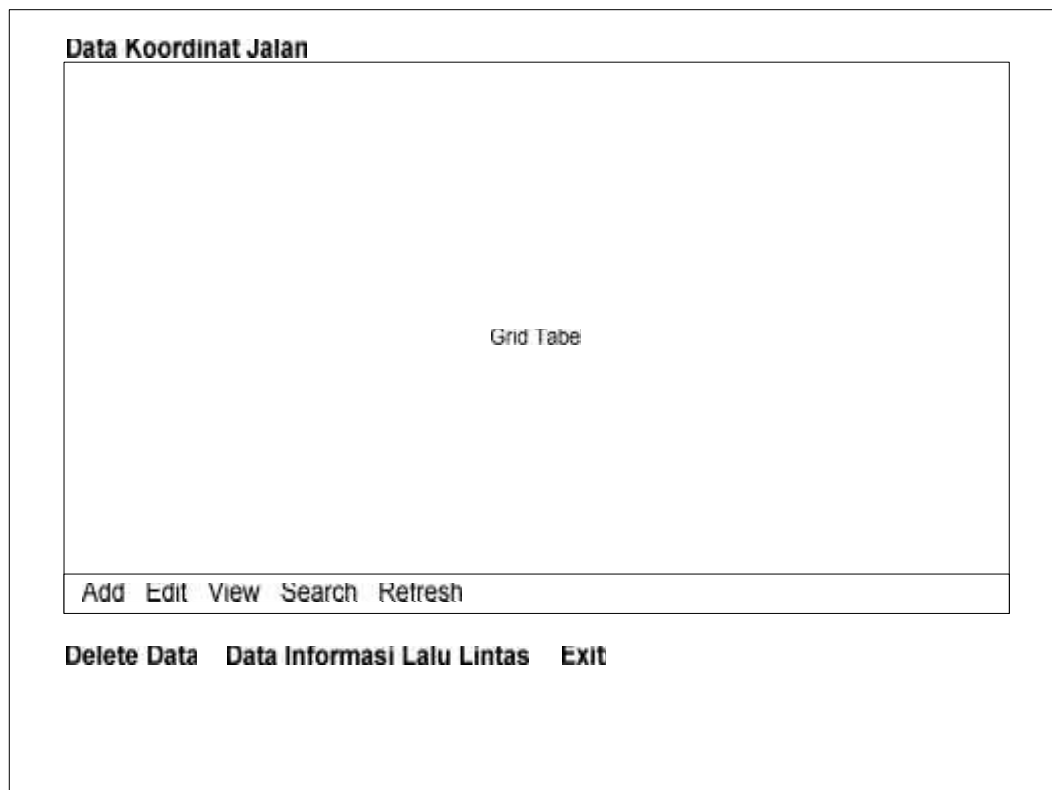
Pada gambar 4.10 hingga gambar 4.12 berikut ini, dijelaskan mengenai rancangan antarmuka *database* aplikasi beserta deskripsinya.

The diagram illustrates a login interface within a large rectangular frame. At the top center is a box labeled "Header". Below it are two rows of input fields. The first row consists of a box labeled "Username" followed by an empty text input field. The second row consists of a box labeled "Password" followed by an empty text input field. Below these fields is a single button labeled "Login".

Gambar 4.10. Antarmuka Login *Database Aplikasi*

Tabel 4.10. Deskripsi antarmuka *Login Database Aplikasi*

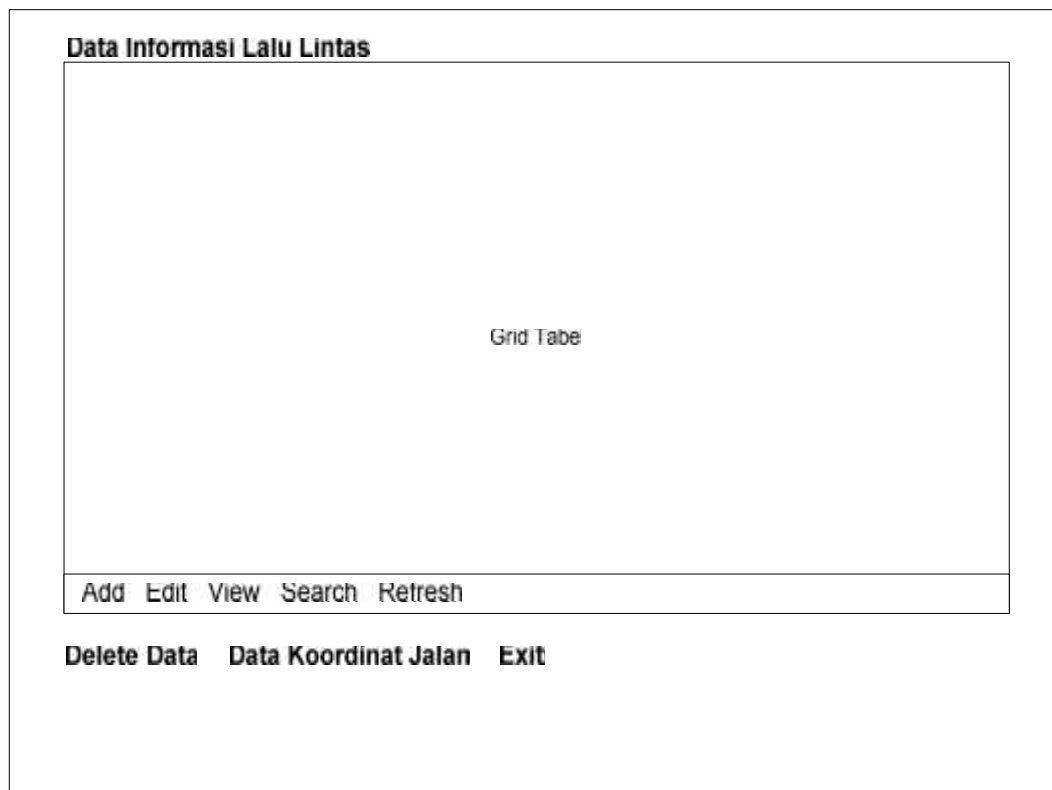
No	Nama <i>Item</i>	Deskripsi
1.	Header	Merupakan Kepala dari antarmuka aplikasi yang menunjukkan karakter dari antarmuka
2.	<i>Username</i>	Merupakan sebuah text view
3.	Password	Merupakan sebuah text view



Gambar 4.11. Antarmuka Data Koordinat Jalan *Database* Aplikasi

Tabel 4.11. Deskripsi antarmuka Data Koordinat Jalan *Database* Aplikasi

No	Nama Item	Deskripsi
1.	Grid Tabel	Merupakan sebuah grid yang akan menampilkan kolom dan baris pada tabel
2.	<i>Add, Edit, View, Search, Delete Data</i>	Merupakan fitur yang bisa digunakan untuk melakukan penambahan, perubahan, melihat, pencarian, dan penghapusan data
3.	Data Informasi Lalu Lintas	Merupakan sebuah link untuk berpindah ke tabel berita



Gambar 4.12. Antarmuka Data Informasi Lalu Lintas *Database* Aplikasi

Tabel 4.12. Deskripsi antarmuka Data Informasi Lalu Lintas *Database* Aplikasi

No	Nama Item	Deskripsi
1.	Grid Tabel	Merupakan sebuah grid yang akan menampilkan kolom dan baris pada tabel
2.	<i>Add, Edit, View, Search, Delete Data</i>	Merupakan fitur yang bisa digunakan untuk melakukan penambahan, perubahan, melihat, pencarian, dan penghapusan data
3.	Data Koordinat Jalan	Merupakan sebuah link untuk berpindah ke tabel koordinat

BAB V

IMPLEMENTASI DAN PENGUJIAN

Bab ini merupakan bagian dari tahapan pembuatan aplikasi dan tahapan *testing* dan pengujian aplikasi, dimana telah dilakukan pengkodean aplikasi, dan akan dilakukan implementasi aplikasi sebagai bagian dari tahapan pembuatan aplikasi, kemudian akan dilakukan pengujian fungsi-fungsi aplikasi dengan metode *Blackbox* dan akan dilakukan pengamatan dari hasil pengujian tersebut untuk mengetahui kekurangan aplikasi dan kemudian dilakukan pengambilan kesimpulan sebagai bagian dari tahapan *testing* dan pengujian aplikasi.

5.1. Implementasi

Tahapan ini merupakan tahap dimana aplikasi akan dibuat setelah melakukan analisa dan perancangan pada bab sebelumnya. Implementasi ini mencakup pembuatan aplikasi, pembuatan *user interface* untuk admin *database*, pembuatan *database* aplikasi, serta penerapan dari algoritma *Dijkstra*.

5.1.1. Pembuatan Aplikasi

5.1.1.1. Pengkodean

Pada tahap ini, akan dibagi menjadi 2 tahap, yaitu tahap pembuatan aplikasi dan tahap implementasi.

A. Pembuatan Aplikasi

Tahapan ini merupakan tahapan dimana aplikasi yang telah dirancang, dianalisa, akan dibangun, lalu diuji kelayakannya untuk selanjutnya dioperasikan sebagaimana mestinya sesuai dengan fungsi dan kelayakannya. Berikut ini akan dijelaskan beberapa hal yang berhubungan dengan pembuatan aplikasi berdasarkan analisis dan perancangan yang telah dilakukan terhadap aplikasi Pekanbaru Taksi Guide berbasis *client-server* pada sistem operasi Android ini.

1. Lingkungan Pengembangan

Perangkat keras dan perangkat lunak merupakan hal yang selalu dibutuhkan untuk mengimplementasikan rancangan yang telah dibuat sebelumnya. Berikut ini akan dijelaskan mengenai lingkungan implementasi yang meliputi lingkungan pada perangkat keras dan perangkat lunak.

a. Spesifikasi Perangkat Keras

Berdasarkan rancangan yang ada, maka untuk membuat aplikasi PTG dibutuhkan beberapa perangkat keras. beberapa perangkat keras pendukung tersebut antara lain adalah:

1. Telepon Genggam (*Handphone*)

Perangkat ini digunakan untuk menjalankan aplikasi PTG dari sisi pengguna. Perangkat yang digunakan yaitu *Samsung Galaxy Gio* dengan spesifikasi:

- a. Sistem Operasi : Android versi 2.2 (*Frozen Yogurt*)
- b. CPU : 600 MHz *Processor*
- c. Ruang Penyimpanan *external* : 1 GB
- d. *Memory* : 158 MB *internal*, 278 MB RAM
- e. Dimensi Layar : 320 x 480 px
- f. Konektivitas : GSM 850 / 900 / 1800 / 1900 & HSDPA 900 / 2100

2. 1 unit komputer sebagai *server database* dengan spesifikasi sebagai berikut:

- a. *Processor* : Intel Xeon (8 *core*, 2,4 GHz)
- b. *Memory* : 10 GB
- c. *Hardisk* : 320 GB SATA

b. Spesifikasi Perangkat Lunak

Dalam membangun aplikasi PTG ini dibutuhkan beberapa perangkat lunak sebagai berikut:

- a. Sistem Operasi

Sistem operasi yang digunakan untuk pemrograman adalah *Windows 7 Ultimate 64 bit*. Sedangkan sistem operasi untuk *server database* adalah *Linux CentOS 5.5 Final 32 bit*.

b. Bahasa Pemrograman

Dalam hal ini yang digunakan adalah bahasa pemrograman *Java* dan *PHP*.

c. *Tools* Pengembangan

Dalam hal ini beberapa *tools* yang digunakan adalah:

1. *Tools* Pengembangan : *Eclipse Galileo 3.5, Notepad ++*
: *Java Development Kit 6u24 (JDK 6u24)*
: *Android SDK, ADT 8.0*
: *Android Virtual Device 2.2 (Froyo) with Google API's 8*

2. *Server* : *PHP, MySQL, PhpMyAdmin*

e. Pemodelan UML : *Argo UML, Microsoft Visio, Rational Rose*

B. Batasan Dalam Pembuatan Aplikasi

Batasan dalam pembuatan aplikasi PTG ini pada tugas akhir ini adalah:

1. Menggunakan bahasa pemrograman *Java* sebagai bahasa pemrograman untuk mengembangkan aplikasi, yang didukung dengan aplikasi *MySQL* bahasa pemrograman *PHP*, serta aplikasi *PhpMyAdmin* sebagai *interface* untuk mengelola *database* pada *server*.
2. Data yang digunakan dalam implementasi ini adalah berupa data simulasi lapangan yang berasal dari *google map* yang telah didigitasi sesuai dengan kebutuhan aplikasi.
3. Aplikasi ini tidak mengembangkan dari sisi *website* sebagaimana aplikasi-aplikasi pada umumnya.

4. Tahap pembuatan aplikasi menggunakan *IDE Eclipse Helios*, *Android SDK version 8* dengan dukungan *Google APIs 8* dan perangkat *Samsung Galaxy Gio*.

5.1.1.2. Implementasi

Tahapan implementasi pada aplikasi ini terbagi menjadi beberapa tahapan. Yaitu hasil implementasi perhitungan algoritma yang digunakan, hasil pengujian algoritma yang digunakan pada beberapa kasus tertentu, dan hasil implementasi pada aplikasi.

A. Hasil Implementasi Perhitungan Algoritma *Dijkstra*

Algoritma *Dijkstra* diterapkan pada graf berarah dan berbobot yang seluruh data dari graf nya telah direpresentasikan ke dalam bentuk tabel. Untuk itu diperlukan data tabel hasil dari representasi titik-titik koordinat jalan yang ada di kota Pekanbaru.

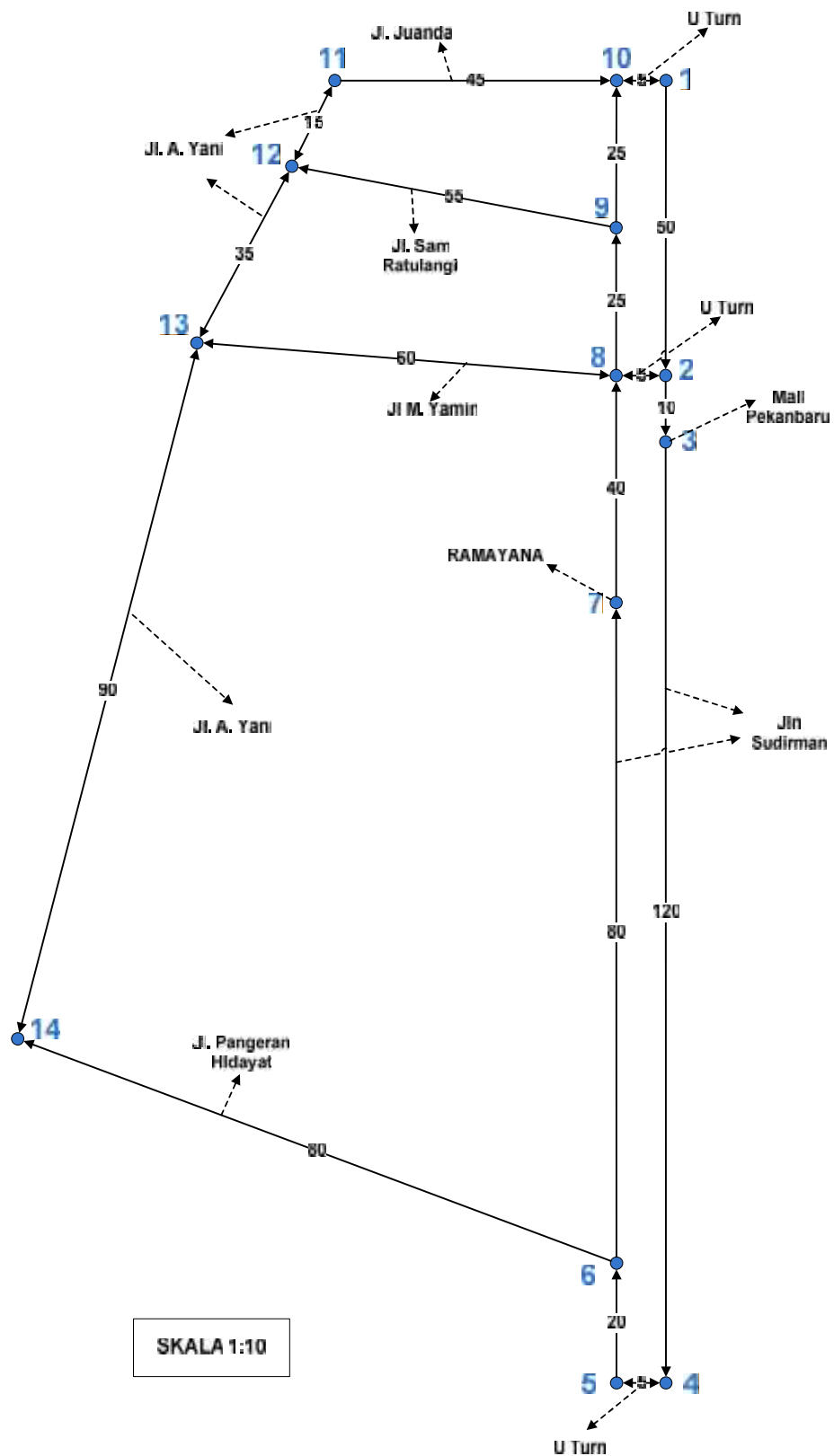
1. Cara Kerja Algoritma *Dijkstra*

Algoritma ini mencari panjang lintasan terpendek dari node asal ke node tujuan dalam sebuah graf. Langkah-langkah dalam menentukan lintasan terpendek pada algoritma *Dijkstra* yaitu :

1. Pada awalnya inisialisasikan node asal (V1) dan node tujuan (V2).
2. Buat 2 buah *list*, *open list* dan *closed list*. Keduanya tidak ada data atau kosong, dan formatnya {node cabang - total bobot - node induk}.
3. Masukkan V1 ke *open list*.
4. Pilih 1 node dengan bobot terkecil pada *open list*, kemudian pindahkan ke dalam *closed list*. Sehingga bobot terkecil tadi tidak terdapat lagi di *open list*.
5. Cari node yang bertetangga langsung dari node sebelumnya, yang masuk terakhir ke dalam *closed list*. Tambahkan bobot dengan node yang terkait, apabila sudah ada dalam *closed list* abaikan.

6. Apabila dalam *open list* terdapat node yang sudah dibandingkan, lalu cari yang terkecil, dan perbaharui. Bila ternyata jumlah bobotnya sama dalam node yang sama, maka abaikan.
7. Apakah data dalam *open list* kosong ? jika belum ulangi langkah 4.
8. Pencarian berhenti apabila node tujuan telah terdapat di *closed list*.
9. Dalam *closed list* cari V2, telusuri jalur berdasarkan node induk sampai mengacu ke node asal (V1), dan balikkan urutan node. Pilih total bobot yg terkecil apabila ada node cabang yang lebih dari 1.
10. Lintasan terpendek ditemukan bersama bobotnya.
11. Algoritma *Dijkstra* digunakan untuk pencarian jalur terpendek dari suatu graf, sehingga akan didapatkan rute tempuh yang terpendek.

Pada Gambar 5.1 dibawah ini adalah contoh dari jalan yang ada di Kota Pekanbaru yang telah dirubah ke dalam bentuk graf berarah dan berbobot sesuai dengan kondisi arah lalu lintas jalan tersebut.



Gambar 5.1. Contoh Graf Jalan Kota Pekanbaru

Tabel 5.1 dibawah ini adalah hasil representasi dari graf jalan pada gambar 5.1 diatas. Tabel inilah yang nantinya yang akan digunakan oleh algoritma *Dijkstra* untuk mencari rute terpendek dari suatu node ke node yang lainnya.

Tabel 5.1. Graf *Dijkstra*

No	NamaJalan	Titik Awal	Titik Akhir	Panjang (m)	Kategori
1	Jl. Sudirman	1	2	50	Jalan
2	Mall Pekanbaru	2	3	10	Mall
3	Jl. Sudirman	3	4	120	Jalan
4	U Turn	4	5	5	U turn
5	U Turn	5	4	5	U turn
6	Jl. Sudirman	5	6	20	Jalan
7	Ramayana	6	7	80	Mall
8	Jl P. Hidayat	6	14	80	Jalan
9	Jl. Sudirman	7	8	40	Jalan
10	Jl. Sudirman	8	9	25	Jalan
11	Jl. Sudirman	9	10	25	Jalan
12	U Turn	10	1	5	U turn
13	U Turn	1	10	5	U turn
14	Jl A. Yani	14	13	90	Jalan
15	Jl A. Yani	13	14	90	Jalan
16	Jl A. Yani	13	12	35	Jalan
17	Jl A. Yani	12	13	35	Jalan
18	Jl A. Yani	11	12	15	Jalan
19	Jl A. Yani	12	11	15	Jalan
20	Jl. Juanda	11	10	45	Jalan
21	Jl. Sam Ratulangi	9	12	55	Jalan
22	Jl. M. Yamin	8	13	60	Jalan
23	Jl. M. Yamin	13	8	60	Jalan
23	U Turn	8	2	5	U turn
25	U Turn	2	8	5	U turn

2. Contoh Perhitungan Pencarian Algoritma *Dijkstra*

Berpedoman pada Gambar 5.1 diatas serta pada Tabel 5.1, maka akan dilakukan proses pencarian yang berasal dari node 14 menuju node 7. Berikut tahapan dalam menghitung rute terpendek menggunakan algoritma *Dijkstra*.

a. Inisialisasi

Tabel 5.2. Inisialisasi

OPEN LIST	<i>CLOSED LIST</i>
(cabang-total bobot-induk)	(induk-total bobot-cabang)
14-0-null	----

Inisialisasi titik awal ke dalam *open list*. *Closed list* kosong.

b. Iterasi 1

Tabel 5.3 Iterasi 1

OPEN LIST	<i>CLOSED LIST</i>
(cabang-total bobot-induk)	(induk-total bobot-cabang)
13-90-14	14-0-null

Iterasi 1, memilih total bobot terkecil pada *open list* dari iterasi sebelumnya, kemudian pindahkan ke *closed list*. dalam hal ini, **14-0-null** adalah bobot terkecil dari *open list* pada iterasi sebelumnya, dan dimasukkan ke *closed list*. Kemudian, buka *open list* yang node induknya adalah node 14 berdasarkan graf atau tabel.

c. Iterasi 2

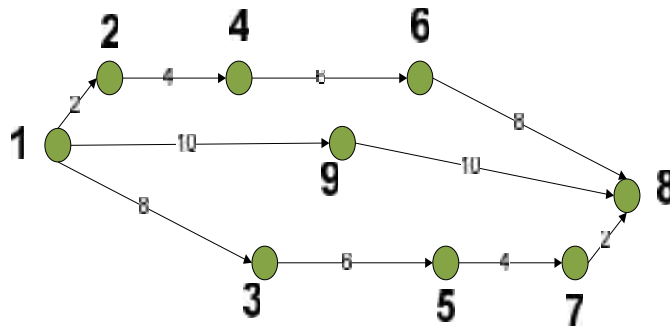
Tabel 5.4 Iterasi 2

OPEN LIST	<i>CLOSED LIST</i>
(cabang-total bobot-induk)	(induk-total bobot-cabang)
14-180-13	14-0-null
8-150-13	13-90-14
12-125-13	

Iterasi 2, memilih total bobot terkecil pada *open list* dari iterasi sebelumnya, kemudian pindahkan ke *closed list*. dalam hal ini, **13-90-14** adalah bobot terkecil dari *open list* pada iterasi sebelumnya, dan dimasukkan ke *closed list*. Kemudian, buka *open list* yang node induknya adalah node 13 berdasarkan graf atau tabel. Untuk iterasi selanjutnya dapat dilihat pada Lamiran C.

B. Hasil Pengujian Algoritma Dijkstra Pada Beberapa Bentuk Graf

1. Graf Tipe A



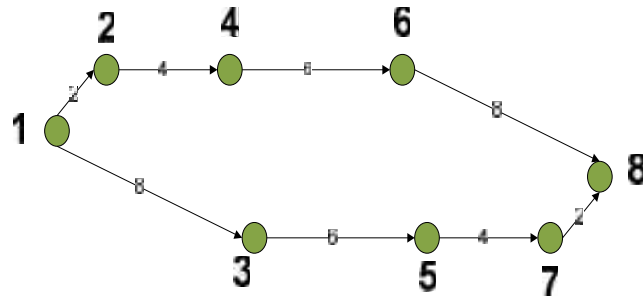
Gambar 5.2. Graf Tipe A

Tabel 5.5 Hasil Representasi Graf Tipe A

No	Node Asal	Node Tujuan	Panjang
1	1	2	2
2	1	3	8
3	2	4	4
4	3	5	6
5	4	6	6
6	5	7	4
7	6	8	8
8	7	8	2
9	1	9	10
10	9	8	10

Pada kondisi graf tipe A, setelah dilakukan pencarian menggunakan algoritma *Dijkstra* dari node asal 1 menuju node tujuan 8, maka didapatkan hasil rute terpendeknya adalah **1 --> 9 --> 8** dengan total bobot adalah **20**.

2. Graf Tipe B



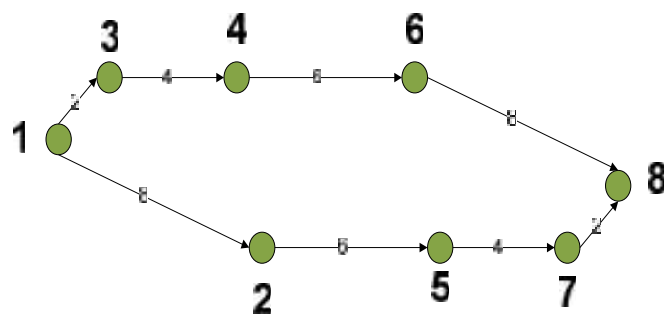
Gambar 5.3. Graf Tipe B

Tabel 5.6 Hasil Representasi Graf Tipe B

No	Node Asal	Node Tujuan	Panjang
1	1	2	2
2	1	3	8
3	2	4	4
4	3	5	6
5	4	6	6
6	5	7	4
7	6	8	8
8	7	8	2

Pada kondisi graf tipe B, setelah dilakukan pencarian menggunakan algoritma *Dijkstra* dari node asal 1 menuju node tujuan 8, maka didapatkan hasil rute terpendeknya adalah 1--> 2 --> 4 --> 6 --> 8 dengan total bobot adalah 20.

3. Graf Tipe C



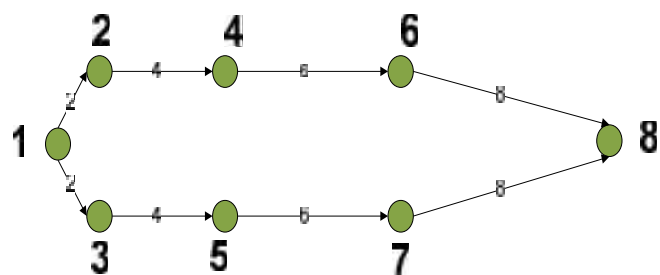
Gambar 5.4. Graf Tipe C

Tabel 5.7 Hasil Representasi Graf Tipe C

No	Node Asal	Node Tujuan	Panjang
1	1	3	2
2	1	2	8
3	3	4	4
4	2	5	6
5	4	6	6
6	5	7	4
7	6	8	8
8	7	8	2

Pada kondisi graf tipe C, setelah dilakukan pencarian menggunakan algoritma *Dijkstra* dari node asal 1 menuju node tujuan 8, maka didapatkan hasil rute terpendeknya adalah **1 --> 3 --> 4 --> 6 --> 8** dengan total bobot adalah **20**.

4. Graf Tipe D



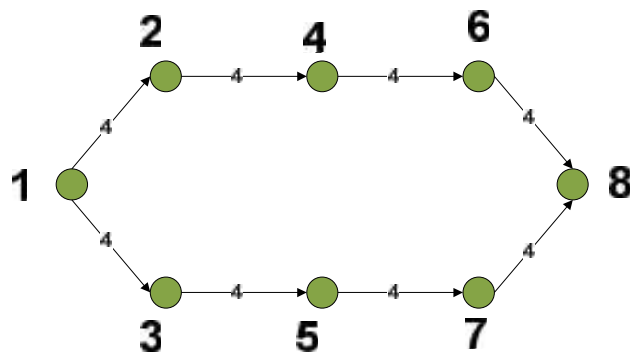
Gambar 5.5. Graf Tipe D

Tabel 5.8 Hasil Representasi Graf Tipe D

No	Noda Asal	Node Tujuan	Panjang
1	1	2	2
2	1	3	2
3	2	4	4
4	3	5	4
5	4	6	6
6	5	7	6
7	6	8	8
8	7	8	8

Pada kondisi graf tipe D, setelah dilakukan pencarian menggunakan algoritma *Dijkstra* dari node asal 1 menuju node tujuan 8, maka didapatkan hasil rute terpendeknya adalah **1 --> 2 --> 4 --> 6 --> 8** dengan total bobot adalah **20**.

5. Graf Tipe E



Gambar 5.6. Graf Tipe E

Tabel 5.9 Hasil Representasi Graf Tipe E

No	Node Asal	Node Tujuan	Panjang
1	1	2	4
2	1	3	4
3	2	4	4
4	3	5	4
5	4	6	4
6	5	7	4
7	6	8	4
8	7	8	4

Pada kondisi graf tipe E, setelah dilakukan pencarian menggunakan algoritma *Dijkstra* dari node asal 1 menuju node tujuan 8, maka didapatkan hasil rute terpendeknya adalah **1 --> 2 --> 4 --> 6 --> 8** dengan total bobot adalah **20**.

C. Implementasi Fitur Pemberitahuan (*Alert*) Aplikasi

Fitur *alert* berfungsi sebagai pemberitahuan kepada pengguna aplikasi apabila koordinat pengguna tidak berada pada rute perjalanan yang telah ditentukan sebelumnya.

1. Komponen Fitur *Alert*

Dalam penerapannya, fitur ini membutuhkan beberapa komponen seperti perangkat GPS, koneksi GPRS, dan *server*.

a. GPS

Dalam menerapkan fitur ini, *gps* sangat berperan dalam menangkap koordinat pengguna. Koordinat pengguna dikirim ke *server* untuk terus dipantau. Untuk itu dibutuhkan sebuah perangkat GPS yang mampu mengirimkan serta menerima sinyal dari satelit dengan cepat. Dan mampu menangkap koordinat pengguna ketika pengguna sedang bergerak.

b. Koneksi GPRS

Koordinat yang didapatkan oleh GPS, akan dikirimkan ke *server* untuk diolah melalui perangkat *mobile* pengguna. Untuk itu dibutuhkan koneksi GPRS yang memiliki respon dan koneksi yang cepat dalam pengiriman dan penerimaan paket data.

c. *Server*

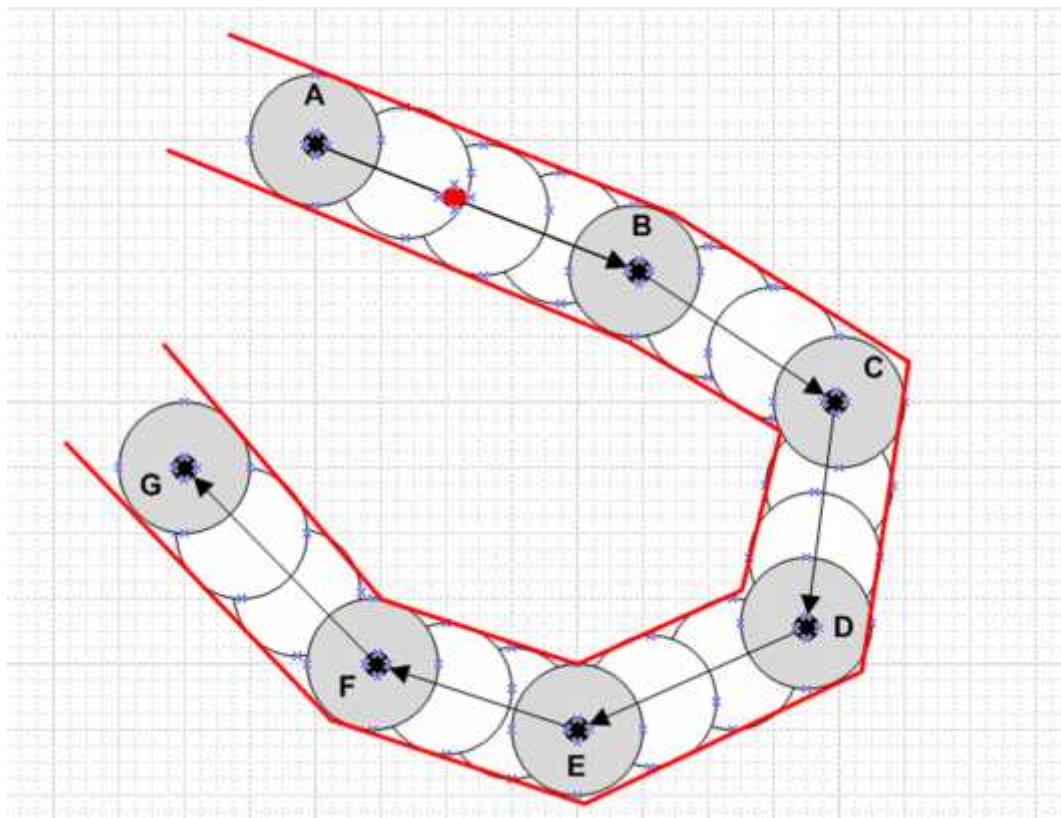
Selain untuk mendapatkan rute terpendek, *server* juga mengolah data koordinat pengguna untuk dilakukan penyesuaian dengan data rute terpendek yang akan ditempuh. Apabila hasil pengolahan menunjukkan bahwa koordinat pengguna tidak berada pada jalur yang telah ditetapkan, maka *server* akan mengirimkan informasi kepada perangkat pengguna dan kemudian *alert* akan aktif.

2. Proses Kerja *Alert*

Dalam menentukan apakah koordinat pengguna yang diterima *server* berada pada jalur yang telah ditentukan sebelumnya atau tidak, maka dibutuhkan sebuah perhitungan yang mampu memberikan keputusan apakah pengguna berada dijalurnya atau tidak.

Perhitungan ini yakni, memulai dengan membuat daerah *range/cakupan* koordinat dari setiap titik koordinat jalan yang telah ditandai oleh di peta. Kemudian membandingkan apakah koordinat pengguna masih berada di daerah *range/cakupan* koordinat atau tidak.

Pada gambar 5.7 dibawah ini menjelaskan detail dari daerah *range/cakupan* koordinat sebagai batasan koordinat pengguna.



Gambar 5.7. Daerah *Range/Cakupan* Koordinat

Keterangan gambar:

1. Titik hitam A, B, C, D, E, F, G adalah titik koordinat jalan yang ditandai dipeta.
2. Garis panah berwarna hitam yang menghubungkan antar titik berwarna hitam merupakan arah tujuan yang harus diikuti pengguna. Garis ini merupakan sebuah rute terpendek untuk menuju lokasi tujuan.
3. Lingkaran berwarna abu-abu merupakan daerah *range*/cakupan koordinat yang telah dihitung besarnya.
4. Lingkaran berwarna putih merupakan daerah *range*/cakupan koordinat semu yang saling menghubungkan antara lingkaran abu-abu.
5. Titik berwarna merah merupakan koordinat pengguna.
6. Garis berwarna merah merupakan pinggir jalan. Ini juga menandai seberapa lebar jalan yang bisa dilalui.

Setelah rute terpendek didapat melalui perhitungan algoritma *Dijkstra* yang dilakukan *server*, maka akan ditentukan hasil perhitungan daerah *range*/cakupan koordinat sebagai batas koordinat yang tidak boleh dilalui pengguna hingga sampai ketujuan. Daerah *range*/cakupan koordinat ini juga termasuk daerah *range*/cakupan koordinat semu (pada gambar 5.7 adalah lingkaran yang berwarna putih). Tujuannya agar tidak terjadi putusnya daerah *range*/cakupan koordinat. Daerah ini adalah batasan yang boleh dilalui oleh pengguna.

Kemudian, aplikasi menangkap koordinat pengguna melalui perangkat GPS, lalu mengirimkan koordinat tersebut ke *server* untuk dilakukan pendeteksian posisi pengguna berdasarkan data daerah *range*/cakupan koordinat yang telah didapat. Apabila koordinat pengguna masih berada di daerah tersebut, maka *server* mengirimkan informasi bahwa koordinat pengguna masih berada di daerah *range*/cakupan koordinat dan *alert* tidak aktif. Sebaliknya, jika koordinat pengguna tidak berada di daerah *range*/cakupan koordinat, maka *server* mengirimkan informasi ke pengguna bahwa koordinat pengguna tidak berada

pada daerah *range*/cakupan koordinat, dan *alert* akan aktif sehingga aplikasi memberitahukan informasi kepada pengguna bahwa posisi pengguna sedang tidak berada pada rute yang telah ditentukan sebelumnya.

Proses ini terus berlangsung hingga koordinat pengguna telah berada di daerah *range*/cakupan koordinat tujuan.

D. Hasil Implementasi

1. Hasil Implementasi Aplikasi

Pada tahapan hasil implementasi ini, seluruh *package source* dirubah menjadi sebuah file yang berformat .apk yang diberi nama *taksiguide.apk* yang telah di install ke perangkat *Samsung Galaxy Gio*.

Secara garis besar, seluruh antarmuka bisa diakses dengan menyentuh layar (*tap*) pada perangkat, kecuali untuk membuka menu aplikasi dengan menekan tombol menu pada perangkat (tidak menyentuh bagian layar aplikasi).

Terdapat beberapa antarmuka yang digunakan untuk interaksi antara pengguna dengan PTG. Antarmuka tersebut dijelaskan sebagai berikut:

1. Tampilan *Opening* Aplikasi.

Merupakan tampilan awal yang dilihat pengguna ketika membuka aplikasi. Pada tampilan *opening* ini, diberi suara sebagai tanda bahwa aplikasi terbuka yang disertai dengan gambar taksi yang muncul.

2. Tampilan *Home*

Adalah tampilan utama dari aplikasi PTG. pada tampilan ini, terdapat 1 menu yang bisa di akses dengan cara *tapping*, yaitu menu Nomor Kontak Taksi. ketika pengguna menekan tombol menu pada perangkat, maka akan muncul 5 menu, yaitu Tujuan, Lihat Peta, Pencarian, Tentang Apps, dan Keluar.

3. Menu Nomor Kontak Taksi

Menu ini berisi nomor-nomor kontak taksi yang ada di Kota Pekanbaru.

4. Menu Tujuan

Menu ini menuntun pengguna untuk memilih kategori tujuan yang diinginkan, memilih lokasi tujuan, serta menunjukkan rute terpendek untuk bisa mencapai lokasi yang diinginkan.

5. Menu Lihat Peta

Menu ini menampilkan seluruh lokasi tujuan yang terdapat di *database* yang ditandai dengan gambar atau *icon* tertentu berdasarkan kategorinya.

6. Menu Pencarian

Pada menu ini, pengguna bisa melakukan ke seluruh lokasi tujuan dengan mengetik lokasi yang diinginkan pada *field* yang telah disediakan. Ketika lokasi tujuan didapat, pengguna bisa langsung diarahkan menuju lokasi tujuan.

7. Menu Tentang Apps

Menu ini berisi mengenai informasi dari aplikasi PTG.

Tampilan antarmuka utama atau Home pada aplikasi PTG dapat dilihat pada Gambar 5.2 dibawah ini.



Gambar 5.7. Tampilan *Home* Aplikasi



Gambar 5.8. Tampilan Ketika Tombol Menu di perangkat ditekan (Menu *Home*)

Untuk hasil implementasi yang lebih lengkap mengenai aplikasi PTG dapat dilihat pada Lampiran D.

2. Hasil Implementasi Antarmuka *Database* Aplikasi

Pada tahap implementasi ini, seluruh *connector* php telah berada di *server* dan siap untuk diakses melalui alamat url 180.250.43.124/taksi/admin/login.php.

Gambar 5.8 hingga 5.9 dibawah ini merupakan tampilan dari antarmuka *database* aplikasi.



Gambar 5.9. Tampilan Login *Database* Aplikasi

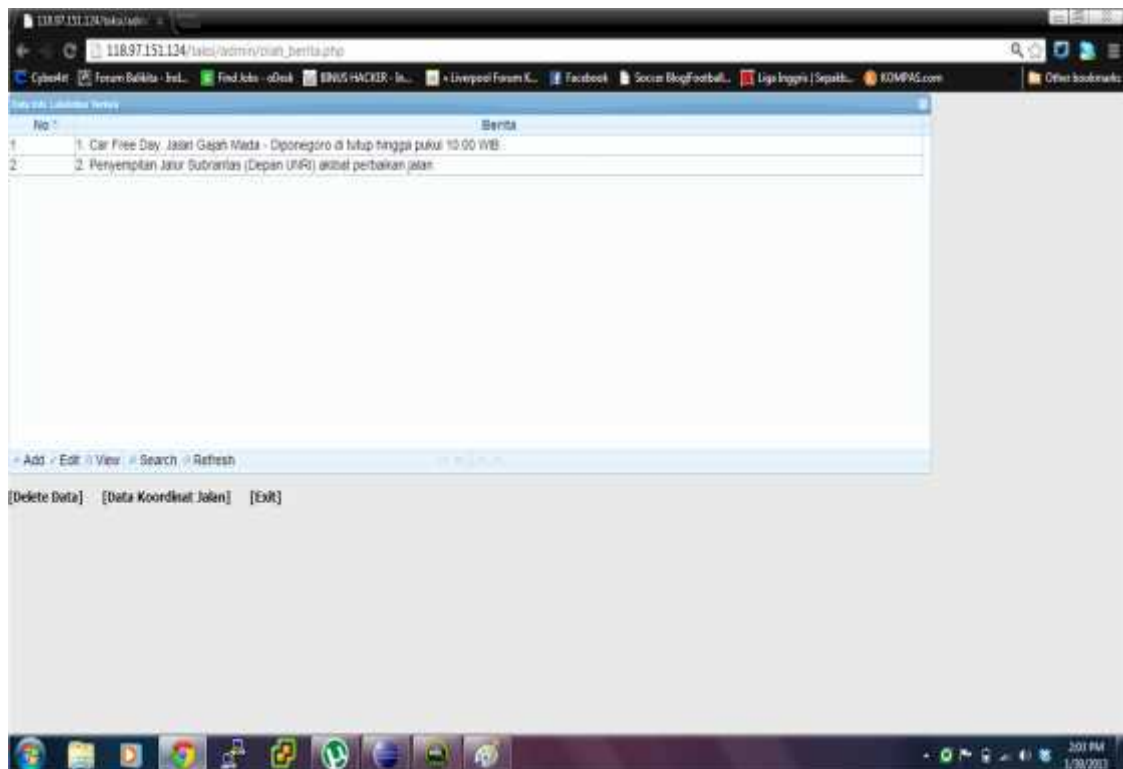
Do you want Google Chrome to save your password? Save password Never for this site

No	Nama Jalas	Titik Awal	Titik Akhir	X1	Y1	X2	Y2	Panjang	Keterangan	Kategori
1	Uluh SUSKA 1	477		101.361813	0.46554	101.363349	0.46215	414	Dari Uluh ke Lintas	
2	Lintas Peka 477	478		101.363349	0.46215	101.367318	0.464307	500	Tum ngiri lintas p	
3	Lintas Peka 478	2		101.367318	0.464307	101.368265	0.464301	110	simpang 4 panam	
4	Uluh SUSKA 1	102		101.361813	0.46554	101.366611	0.468968	660	Dari Uluh ke Garuc	
5	Jl. HR. Subir 2	3		101.368265	0.464301	101.368462	0.464328	22	Point 2.3. HR. Su	
6	Jl. HR. Subir 3	4		101.368462	0.464328	101.371158	0.464307	297	U Tum 15-Jl. HR	
7	Jl. HR. Subir 4	5		101.371158	0.464307	101.371152	0.464224	10	Value U Tum 15 c	
8	Jl. HR. Subir 5	4		101.371152	0.464224	101.371158	0.464307	10	Value U Tum 15 c	
9	Jl. HR. Subir 4	6		101.371158	0.464307	101.375192	0.464283	440	U Tum 16-Jl. HR	
10	Jl. HR. Subir 6	7		101.375192	0.464283	101.375194	0.464109	10	Value U Tum 16 c	
11	Jl. HR. Subir 7	6		101.375194	0.464109	101.375192	0.464283	10	Value U Tum 16 c	
12	Jl. HR. Subir 6	8		101.375192	0.464283	101.378107	0.464264	320	U Tum 17-Jl. HR	
13	Jl. HR. Subir 8	9		101.378107	0.464264	101.378105	0.464181	10	Value U Tum 17 c	
14	Jl. HR. Subir 9	8		101.378105	0.464181	101.378107	0.464264	10	Value U Tum 17 c	
15	Jl. HR. Subir 8	10		101.378107	0.464264	101.379014	0.464261	101	U Tum 18-Jl. HR	
16	Jl. HR. Subir 10	11		101.379014	0.464261	101.379011	0.464175	10	Value U Tum 18 c	
17	Jl. HR. Subir 11	10		101.379011	0.464175	101.379014	0.464261	10	Value U Tum 18 c	
18	Jl. HR. Subir 10	137		101.379014	0.464261	101.382125	0.464234	346	Point Depan RS J	
19	RS JIWA PAJAT 137	138		101.382125	0.464234	101.382109	0.464877	51	RS JIWA PANANJ rutan sat	
20	RS JIWA PAJAT 138			101.382109	0.464877	101.382109	0.464877	0		

ADD Edit View Search Refresh

[Delete data] [Data Informasi Lahu-Lintas] [Edit]

Gambar 5.10. Tampilan Data Tabel Koordinat Jalan



Gambar 5.11. Tampilan Data Tabel Informasi Lalu Lintas

5.2. Pengujian

Tahapan pengujian merupakan tahapan dimana akan dilakukan pengujian terhadap aplikasi. Pengujian dilakukan dengan metode pengujian blackbox. Selain itu, dilakukan juga pengujian terhadap akses ke aplikasi pada lokasi tertentu, jam tertentu, serta menggunakan provider internet tertentu.

5.2.1. Testing dan Pengujian Aplikasi

Tahapan ini dilakukan untuk mengetahui apakah sistem yang dibangun telah sesuai dengan yang diharapkan. Tujuan utama dari *testing* dan pengujian sistem adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan. Salah satu metode pengujian jenis ini dikenal dengan pengujian *blackbox*.

A. Pengujian *Blackbox* Aplikasi *Pekanbaru Taksi Guide*

Pada tahap pengujian sistem ini, perangkat keras yang digunakan yaitu *smartphone* Samsung Galaxy Gio. Sedangkan material pengujian untuk sistem ini

menggunakan data koordinat yang telah dimasukan oleh Admin *Database* ke *database* pada *server*. Pengujian yang akan dilakukan adalah pengujian akses ke sistem dan pengujian fungsionalitas sistem menggunakan metode *blackbox*. Hasil dari pengujian ini dapat dilihat pada tabel 5.5.

Tabel 5.10. Hasil Pengujian *Blackbox*

No	Komponen Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Keterangan
1	Halaman Utama Aplikasi	Pengguna mengklik aplikasi di perangkat dan akan muncul opening aplikasi beserta suara pembuka baru kemudian menuju ke <i>home</i> aplikasi	<i>Opening</i> aplikasi berjalan, suara bisa didengar dan <i>home</i> aplikasi berhasil ditampilkan	Benar
2	Menu Tujuan	Pengguna mengklik tombol menu pada perangkat android dan memilih menu Tujuan kemudian akan menampilkan kategori tujuan	Kategori Tujuan berhasil ditampilkan	Benar
3	Daftar Kategori Tujuan	Pengguna memilih kategori yang diinginkan dan akan ditampilkan daftar lokasi tujuannya	Daftar lokasi tujuan berhasil ditampilkan	Benar
4	Memilih 1 daftar lokasi tujuan	Pengguna memilih salah satu dari daftar lokasi tujuan dan akan muncul tampilan Tujuan	Tampilan Tujuan ditampilkan	Benar
5	Button GO !!! pada Tujuan	Pengguna mengklik tombol GO. jika koordinat pengguna sudah didapat oleh GPS	Koordinat didapat dan muncul direction rute terpendek menuju	Benar



		maka akan ditampilkan direction rute terpendek menuju lokasi tujuan. jika tidak, akan ada pemberitahuan bahwa koordinat belum didapat.	lokasi tujuan beserta informasi mengenai perkiraan biaya perjalanan	
6	Menu Lihat Peta	Pengguna menekan tombol menu pada perangkat android dan memilih menu Lihat Peta. akan ditampilkan peta yangtelah ditandai dengan icon	Peta berhasil dimunculkan dengan iconnya	Benar
7	Menu Pencarian	Pengguna menekan tombol menu pada perangkat android dan memilih menu Pencarian. dan akan menampilkan seluruh daftar lokasi tujuan.	Daftar lokasi tujuan berhasil ditampilkan	Benar
8	Menu Nomor Kontak Taksi	Pengguna mengklik tombol Nomor Kontak Taksi pada home aplikasi dan akan menampilkan daftar nomor kontak taksi yang bisa dihubungi.	Daftar nomot kontak taksi berhasil ditampilkan	Benar
9	Menu Tentang Apps	Pengguna menekan tombol menu pada perangkat android dan memilih menu Tentang Apps dan akan menampilkan informasi mengenai aplikasi PTG	Informasi berhasil ditampilkan	Benar



B. Pengujian Akses Aplikasi *Pekanbaru Taksi Guide*



Pengujian yang dilakukan bertujuan untuk mengetahui proses hasil dari sistem, yaitu memperlihatkan aplikasi PTG yang telah dibangun bisa diakses melalui berbagai perangkat Android dan berbagai versi Android. Hasil dari pengujian dapat dilihat halaman *home* telah tampil, dan semua menu serta fitur dapat digunakan.


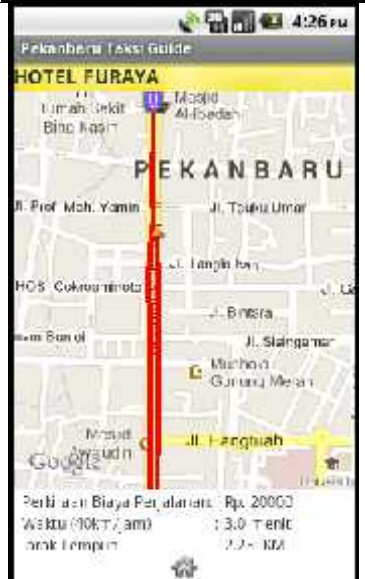
Pada tabel 5.6 di bawah ini menjelaskan pengujian akses ke aplikasi PTG di beberapa lokasi yang berbeda.

Tabel 5.11 Pengujian akses ke aplikasi PTG di beberapa lokasi berbeda dan provider yang berbeda

No	Pukul / Tanggal	Lokasi Awal	Lokasi Tujuan	Provider	Pengujian	Hasil	Tampilan Hasil Pengujian
1	10.00 WIB / 19 November 2012	Kampus UIN (Panam)	Hotel Mona Plaza	Telkomsel	Koordinat pengguna	Berhasil	
					Go	Berhasil	
					Lihat Peta	Berhasil	
					List Lokasi	Berhasil	
					Pencarian	Berhasil	
2	10.25 WIB / 19 November 2012	Kampus UIN (Panam)	Mall SKA Pekanbaru	Telkomsel	Koordinat pengguna	Berhasil	
					Go	Berhasil	
					Lihat Peta	Berhasil	
					List Lokasi	Berhasil	
					Pencarian	Berhasil	

3	11.20 WIB / 19 November 2012	Komplek Kantor Gubernur	Kuliner Bakmi Jakarta Subrantas	XL	Koordinat pengguna	Berhasil	
					Go	Berhasil	
					Lihat Peta	Berhasil	
					List Lokasi	Berhasil	
					Pencarian	Berhasil	
4	11.00 WIB / 19 November 2012	Komplek Kantor Gubernur	Taman Rekreasi Alam Mayang	Telkomsel	Koordinat pengguna	Berhasil	
					Go	Berhasil	
					Lihat Peta	Berhasil	
					List Lokasi	Berhasil	
					Pencarian	Berhasil	

5	11.00 WIB / 20 November 2012	Jl. K.H. Ahmad Dahlan (Kampus UIN Sukajadi)	Rumah Sakit Eka Hospital	Telkomsel	Koordinat pengguna	Berhasil	
					Go	Berhasil	
					Lihat Peta	Berhasil	
					List Lokasi	Berhasil	
					Pencarian	Berhasil	
6	11.30 WIB / 20 November 2012	Jl. K.H. Ahmad Dahlan (Kampus UIN Sukajadi)	Bandara SSQ II	XL	Koordinat pengguna	Berhasil	
					Go	Berhasil	
					Lihat Peta	Berhasil	
					List Lokasi	Berhasil	
					Pencarian	Berhasil	

7	10.30 WIB / 21 Juni 2013	Hotel Furaya	Mall Pekanbaru	XL	Koordinat pengguna	Berhasil	
					Go	Berhasil	
					Lihat Peta	Berhasil	
					List Lokasi	Berhasil	
					Pencarian	Berhasil	
8	10.35 WIB / 21 Juni 2013	Mall Pekanbaru	Hotel Furaya	XL	Koordinat pengguna	Berhasil	
					Go	Berhasil	
					Lihat Peta	Berhasil	
					List Lokasi	Berhasil	
					Pencarian	Berhasil	

Tabel 5.12 Pengujian Aplikasi Pada Beberapa Perangkat dan Sistem Operasi

No	Merk Perangkat	Sistem Operasi	Hasil Pengujian Aplikasi
1	<i>Samsung Galaxy Gio</i>	<i>Android Frozen Yogurt 2.2</i>	Berjalan dengan lancar
2	<i>Sony Experia Myro</i>	<i>Android Ice Cream Sandwich 4.0</i>	Berjalan dengan lancar
3	<i>Samsung Galaxy Ace II</i>	<i>Android Gingerbread 2.3</i>	Berjalan dengan lancar
4	<i>Hisense E860 Smartfren Andro Max</i>	<i>Android Ice Cream Sandwich 4.0</i>	Berjalan dengan lancar
5	<i>Lenovo P770</i>	<i>Android Jelly Bean 4.1</i>	Berjalan dengan lancar

C. Pengujian Fitur Alert

Berdasarkan hasil analisa dari proses kerja fitur ini, maka hingga saat ini fitur *alert* masih belum dapat terealisasi karena disebabkan oleh beberapa faktor. Yaitu:

1. Kemampuan perangkat GPS dalam mendapatkan koordinat pengguna ketika sedang bergerak. Setiap perangkat *mobile* memiliki kemampuan yang berbeda-beda dalam penggunaan GPS karena didukung oleh spesifikasi perangkat itu sendiri seperti prosesor, memory, dan tipe gps yang digunakan.
2. Perhitungan daerah *range*/cakupan koordinat berdasarkan rute terpendek yang telah ditentukan sebelumnya masih belum terpecahkan. Hal ini terjadi karena penentuan daerah *range* ini berhubungan dengan data koordinat *longitude* dan *latitude* yang langsung merepresentasikan objek di muka bumi yang memiliki tingkat akurasi tinggi. Selain itu, penentuan daerah *range*/cakupan koordinat semu sebagai pembantu agar seluruh daerah *range*/cakupan koordinat dapat saling terhubung masih belum ditemukan solusinya.

5.2.2. Kesimpulan Pengujian

Setelah dilakukan beberapa pengujian terhadap aplikasi yang telah dibangun, maka dapat ditarik kesimpulan dari hasil pengujian tersebut. Berikut kesimpulannya:

1. Aplikasi yang dijalankan di beberapa perangkat Android yang berbeda merk, versi Android, dan berbeda operator dapat berjalan dengan lancar.
2. Lamanya pencarian rute terpendek oleh *server* disebabkan oleh beberapa faktor, diantaranya koneksi internet pada perangkat android, serta posisi awal pengguna menuju lokasi tujuan. Semakin jauh dari tujuan maka akan membutuhkan waktu yang tidak sebentar dalam menentukan rute terpendeknya.
3. Penggunaan perangkat GPS untuk menentukan koordinat pada perangkat pengguna sangat dominan. Apabila koordinat pengguna berhasil didapat, maka aplikasi bisa berjalan sebagaimana mestinya. Apabila tidak, maka aplikasi tidak dapat membantu pengguna untuk menentukan *direction* rute terpendek.
4. Rute terpendek yang dilalui dari lokasi awal ke lokasi tujuan belum tentu sama dengan rute terpendek dari lokasi tujuan ke lokasi awal. Hal ini bisa dilihat dari hasil pengujian pada tabel 5.7 point 7 dan 8, ini disebabkan oleh peraturan jalan yang ada di Kota Pekanbaru. Sehingga menghasilkan rute yang berbeda dari sebelumnya dengan jarak tempuh dan waktu tempuh yang berbeda.

BAB VI

PENUTUP

6.1 Kesimpulan

Setelah menyelesaikan serangkaian tahapan dalam merancang dan membangun aplikasi *Pekanbaru Taksi Guide* berbasis *client-server* yang menerapkan teknologi *Location Based Service* dan algoritma *Dijkstra* pada sistem operasi *Android* hingga pada tahapan pengujian, maka dapat diambil beberapa kesimpulan diantaranya adalah sebagai berikut:

1. Kecepatan pencarian algoritma *Dijkstra* berbanding lurus dengan jumlah node yang dihitung. Semakin banyak jumlah node yang dihitung maka kecepatan perhitungan akan semakin lama. Begitu juga sebaliknya.
2. Dalam melakukan *update* data, GUI (*Graphic User Interface*) untuk admin *database* tidak terhubung langsung dengan *Google Map*, sehingga dibutuhkan seorang admin yang mampu menguasai *database* aplikasi untuk bisa melakukan *update* data.
3. Model *database* yang digunakan dalam penelitian ini merupakan model *database* senarai yang merupakan hasil representasi graf jalan Kota Pekanbaru yang mampu menunjukkan posisi geometri pada peta.
4. Ketika didapat rute terpendek dengan total bobot sama yang lebih dari 1, maka algoritma *Dijkstra* akan memilih rute dengan jumlah node paling sedikit dilalui. Jika jumlah node yang dilewati sama, maka algoritma akan memilih rute dengan node pertamanya yang dilewati berada didaftar paling atas dari tabel representasi graf.

6.2 Saran

Beberapa hal yang disarankan dalam pengembangan aplikasi *Pekanbaru Taksi Guide* berbasis *client-server* yang menerapkan teknologi *Location Based*

Service dan algoritma *Dijkstra* pada sistem operasi Android ini adalah sebagai berikut:

1. Untuk pengembangan aplikasi selanjutnya, diharapkan pada data jalan yang digunakan sudah mencakup seluruh tipe jalan dan jalan yang ada di Kota Pekanbaru yang disesuaikan dengan peraturan lalu lintas yang ada.
2. Pada pengembangan aplikasi selanjutnya, diharapkan agar mampu memberikan hasil pencarian yang lebih cepat dengan menambahkan algoritma optimasi dalam melakukan pencarian rute terpendek.
3. Pada pengembangan aplikasi selanjutnya, diharapkan *server database* bisa menyimpan hasil pencarian yang pernah dilakukan sebelumnya, sehingga apabila ada pencarian lokasi yang berasal dari lokasi yang sama menuju lokasi tujuan yang sama pula, maka akan langsung menampilkan hasilnya atau *direction* tanpa perlu melakukan pencarian kembali.
4. Pada pengembangan aplikasi selanjutnya, diharapkan telah memiliki *user interface* untuk admin *database* aplikasi yang telah terhubung langsung dengan *Google Map*, sehingga perubahan dan pengolahan data dapat dilakukan dengan lebih sederhana.
5. Diharapkan pada penelitian berikutnya, proses perhitungan daerah *range/cakupan* koordinat dapat segera terwujud sehingga fitur *alert* pada aplikasi dapat terealisasi.

DAFTAR PUSTAKA

- Adeline Narwastu dan Eri Prasetyo W. *Perancangan Sistem Informasi Geografis Daerah Banjur Di DKI Jakarta Dengan Menggunakan Arc View*. Proceeding PESAT (Psikologi, Ekonomi, Sastra, dan Sipil) Vol. 2, Agustus, 2007.
- Bassil, Youssef. *A Simulation Model for the Waterfall Software Development Life Cycle*. International Journal of Engineering & Technology (iJET), Vol. 2, No. 5, 2012, ISSN: 2049-3444.
- Faizah Ifatul, *Rancang Bangun Perangkat Lunak Penentuan Rute Perjalanan Wisata Di Malang Menggunakan Algoritma Dijkstra*. Laporan Tugas Akhir Sarjana, Jurusan Teknik Informatika, UIN Maulana Malik Ibarahim. Malang, 2010.
- Johnsonbaugh, Richard. *Pearson International Edition : Discrete Mathematics Seventh Edition*. Penerbit Pearson Prentice Hall, United States of America, 2005.
- Kartika Gunadi, Yulia dan Jeffrey Tanuhardja. *Perencanaan Rute Perjalanan Di Jawa Timur Dengan Dukungan GIS Menggunakan Metode Dijkstra's*. Jurnal Informatika Vol. 3, No. 2, Nopember 2002: Hal: 68 – 73.
- Luh Joni Erawati Dewi. *Pencarian Rute Terpendek Tempat Wisata Di Bali Dengan Menggunakan Algoritma Dijkstra*. Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010). Yogyakarta, Juni, 2010. ISSN: 1907-5022.
- Munir, Rinaldi. *Buku Teks Ilmu Komputer : Matematika Diskrit Edisi Ketiga*. Penerbit Informatika, Bandung, 2005
- Nasution H.M.N, *Manajemen Transportasi*, Penerbit Ghalia Indonesia, Surabaya, 1996.
- Neven Vr ek, Goran Bubaš, and Neven Bosilj. *User Acceptance of Location-based Services*. International Journal of Human and Social Sciences 4:2, 2009.
- Rational Team. *“Rational Unified Process : Best Practices for Software Development Teams”*. 2001.
- Safaat, Nazruddin. *Android: Pemrograman Aplikasi Mobile Smartphone dan Tablet PC*. Penerbit Informatika, Bandung, 2011.